

Terese Odéen

# Co-simulation of the propulsion system onboard the icebreaker Oden. Kosimulering av framdriftssystemet ombord isbrytaren Oden.

Master's thesis in Marine Technology

Supervisor: Eilif Pedersen

July 2020



Terese Odéen

**Co-simulation of the propulsion  
system onboard the icebreaker Oden.  
Kosimulering av framdriftssystemet  
ombord isbrytaren Oden.**

Master's thesis in Marine Technology  
Supervisor: Eilif Pedersen  
July 2020

Norwegian University of Science and Technology  
Faculty of Engineering  
Department of Marine Technology



Norwegian University of  
Science and Technology





## **MASTER OF TECHNOLOGY THESIS DEFINITION (30 SP)**

<b>Name of the candidate:</b>	Odéen, Terese
<b>Field of study:</b>	Marine control engineering
<b>Thesis title (Norwegian):</b>	Kosimulering av framdriftssystemet ombord isbrytaren Oden.
<b>Thesis title (English):</b>	Co-simulation of the propulsion system onboard the icebreaker Oden.

### **Background**

For the Icebreaker (IB) Oden we have a structural model and a high-fidelity simulation model of water ice-structure interactions implemented in the Simulator for Arctic Marine Structures (SAMS) numerical simulation platform. This numerical model includes definition of a broken ice environmental domain and the ice actions and action effects on the vessel (and between ice floes and other structures) during icebreaker operation. The SAMS platform also has an interface making it possible to connect it to Simulink models.

The objective of this project is to build a simulation model of IB Oden's mechanical propulsion system and connect it to the SAMS simulator, using Core Simulation Environment (CSE). To be able to connect the simulation models to CSE and perform a co-simulation, FMUs must be used. This makes it possible to include the ice loads in the simulations and investigate how ice loads propagate through the propulsion system onto the engines. One can further look at how to utilize the available power in an efficient way and try out different tactics for energy and emission management.

### **Work description**

1) Perform a background and literature review to provide information and relevant references on:

- 2D-sim and bond graph modelling.
- Icebreaker Oden and its relevant documentation.
- Simulator for Arctic Marine Structures (SAMS) and relevant power system simulation platform(s).
- Power and propulsion systems, incl. relevant power system components and power management functions.
- Co-simulation and relevant tools and functionalities.
- Open Simulation Platform (OSP) and Core Simulation Environment (CSE).

Write a list with abbreviations and definitions of terms and symbols, relevant to the literature study and project report.

2) Familiarize yourself with IB Oden; the vessel itself, and its power and propulsion configuration.

Study the relevant components for the IB Oden power and propulsion systems, incl. main functionality and properties. Present an overview of the relevant data needed for later building your power plant simulation model.

3) Describe CSE by its philosophy, intended use, and main architecture, and its capability for co-simulation connecting a power plant simulation model with the SAMS simulation model. Include a figure showing the main architecture of the co-simulation setup.

- Get an overview and learn how to use CSE for making co-simulation models and connecting simulation models using FMUs and the FMI-standard.
- Discuss what new features and requirements that are important for SAMS to enable it for co-simulation in CSE.

4) Create a framework in CSE for co-simulations, incl. a standardized interface so that the connected FMUs are interchangeable. This will allow for an incremental development of the simulation model.

5) Develop a framework in CSE for connecting the different simulation models.



- 6) Develop a simulation model of the power and propulsion system of IB Oden in 20-sim, MATLAB, or other. Export it as an FMU and connect it to CSE. Perform a co-simulation and discuss the results.
- 7) Connect the ice load model to the simulation model and investigate how the ice loads propagate through the simulation model. Analyze and discuss the results.

### Specifications

Every weekend throughout the project period, the candidate shall send a status email to the supervisor and co-advisors, providing two brief bulleted lists: 1) work done recent week, and 2) work planned to be done next week.

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a logical structure to give a clear exposition of background, problem, design, results, and critical assessments. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed upon. It shall be written in English (preferably US) and contain the elements: Title page, abstract, acknowledgement, project definition, list of symbols and acronyms, table of contents, introduction (project motivation, objectives, scope and delimitations), background/literature review, problem formulation, method, results, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and cause consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis description shall be included after the title page. Computer code, pictures, videos, dataserries, etc., shall be included electronically with the report.

**Start date:** 15 January, 2020

**Due date:** As specified by the administration.

**Supervisor:** Eilif Pedersen

**Co-advisor(s):** Roger Skjetne and Jon Bjørnø

## Preface

This master thesis is the concluding part of my Master's degree at the study program Marine Technology at the Norwegian University of Science and Technology (NTNU).

This thesis has been written during the spring semester 2020 and is finished 31th of July 2020. The topic of the thesis is a combination of several fields that I am passionate about, namely the Arctic, engines and simulation models. Together they formed the project *Co-simulation of the propulsion system onboard the icebreaker Oden*.

This thesis a continuation of the project thesis that I wrote in the autumn semester of 2019. The project thesis was used to explore an older co-simulation tool named Coral, which was the result of the ViProMa project, lasting from 2013 to 2016. Additionally, in the project thesis bond graph and several concepts regarding co-simulation, such as leader-follower structure and the FMI-standard was explored.

In the master thesis, the concepts from the project thesis was further explored as well that a new co-simulation software, Core Simulation Environment, was introduced. Ice loads and the Arctic environment was revisited through literature.

Terese Odéen

Trondheim, July 2020

## Acknowledgement

I would also like to address that this spring, the spring semester of 2020, has been quite different from what I had imagined with the Covid-19 pandemic and university being closed. It has been challenging to work alone from home, without the daily community of the office A1.007 and the rest of the students at Marinteknisk Senter. A considerable part of the semester unfortunately went to cope with the pandemic and learning how once again put all energy into studying.

Firstly, I would like to thank Professor Eilif Pedersen for being my supervisor. He has provided me with useful contacts and information about the topic when needed.

Additionally, I would like to thank my co-supervisor Professor Roger Skjetne.

I would like to thank my co-supervisor Jon Bjørnø for running the simulations in SAMS and providing me with the ice load data for Oden.

I would like to thank Geir Dahler in DNV-GL for the providing the torsional vibration calculations for Oden and responding to all my emails.

I would like to express my gratitude to my amazing team, Anni Yang Christensen, Helene Hogstad Fossum and Kari Renslo Instefjord, which has been helpful when discussing challenges, finding solutions and proofreading.

Lastly, I would like to thank my partner Jie that has been living and working by my side during the Covid-19 pandemic and home quarantine.



## Samandrag

Øvst på den nordlege hemisfæren er Arktis med Nordpolen lokalisert i sentrum av Arktis. Det er det fleire definisjonar som blir brukt for å skildre kvar Arktis er og dei tre vanlegaste er polar sirkelen, tregrensa og  $10^\circ$  isotermin. Eit arktisk klima er prega av lave temperaturar og sjøis. Tre typar skip operera i Arktis, dei er isbrytarar, isgåande skip og isforsterka skip. For å tole is lastene er skrog og framdriftssystemet forsterka. For å betre sikkerheita i Arktis er auka kunnskap ein viktig. Då kan bruk av digital tvilling vere eit nyttig verktøy for å oppnå auka forståing av kreftene som oppstår i interaksjonen mellom is og skrog og korleis kreften frå isen forplantar seg i ombord på skipet.

Ein digital tvilling er ein digital representasjon av sitt fysiske motstykke. Denne teknologien mugelegjer meir fullstendige analysar og simuleringar av komplekse system. Simuleringsmodellen til skipet kan vere samansett av mange ulike simuleringsmodellar i tillegg til sensorar, maskinvare og menneskeleg interaksjon. For å få til kommunikasjonen mellom dei ulike simuleringsmodellane trengst eit kosimuleringsverktøy.

Den svenske isbrytaren Oden er valt som utgangspunkt for å undersøke effekten islaster har på skip og framdriftssystemet ombord. Dette undersøkast ved bruk av kosimuleringseverktøyet Core Simulation Environment, CSE, levert av Open Simulation Platform. Rammeverket utvikla i CSE er slik at tilkopla følgjarar kan bytast utan endringar på sjølve rammeverket. Ein eksisterande simuleringsmodell av Oden i Simulator of Arctic Marine Structures, SAMS, av interaksjonar mellom is og skrog skal koplast til en simuleringsmodell av framdriftssystemet ombord. Simulink og 20-sim blei testa som programvare for utvikling av modellen til framdriftssystemet. Det viste seg at verken SAMS eller Simulink var compatible med CSE. Dermed måtte alle følgjarane utviklast i 20-sim. Sidan SAMS ikkje kunne koplast til CSE, måtte islaster hentast frå ei sjølvstendig simulering i SAMS av at isbrytaren Oden som går med konstant tempo på 3 m/s gjennom eit isfelt brukt i staden.

Kosimuleringsmodellen inneheld feil og kan derfor ikkje brukast til å analysere effekten av islaster på framdriftslinja. Grafane frå kosimuleringa viser at simuleringsmodellen av motorane og girkassen ikkje oppfører seg som forventa, til dømes negativt dreiemoment og rotasjonshastigheiter. Noko av feilen skuldast sannsynlegvis ei algebraisk sløyfe i girkassemodell, som det ikkje var tilstrekkeleg med tid til å løyse på ein tilfredsstillande måte. I framtida kan kosimuleringsmodellen til Oden utviklast på ein betre måte ettersom nyare kosimuleringsprogramvarer gjer det mogleg å bruke den planlagde programvara for følgjarane. Arbeidet med denne avhandlinga gjorde det tydeleg at dette prosjektet er gjennomført på eit for tidleg tidspunkt i utviklingsprosessen av kosimuleringsprogramvara CSE. Dette resulterte i mykje tidkrevjande prøving og feiling som kunne vore unngått dersom prosjektet ble gjennomført då den endelege versjonen av kosimuleringsprogramvara var lansert.

## Abstract

At the top of the northern sphere of the Earth lies the Arctic with the North Pole located at the center. There are several definitions that are used to describe the location of the Arctic and the three most common are the polar circle, the tree line and the  $10^{\circ}$  isotherm. An Arctic climate is characterized by low temperatures and sea ice. Three main types of ships operates in the Arctic, namely icebreakers, ice-going ships and ice-strengthened ships. To withstand ice loads, the hull and propulsion system are reinforced. To improve security in the Arctic, increased knowledge is important. The use of digital twins can increase the understanding of the forces that arise in the interaction between ice and hull and give a deeper understanding on how this affects the propulsion system and other systems board the ship.

A digital twin is a digital representation of its physical counterpart. This technology enables more complete analyzes and simulations of complex systems. The simulation model of the vessel can be composed of many different simulation models in addition to sensors, hardware, and human interaction. In order to achieve communication between the different simulation models, a co-simulation tool is needed.

The Swedish icebreaker Oden has been chosen as a starting point for investigating the effect of ice loads on ships and the propulsion line onboard. This was investigated using the co-simulation tool Core Simulation Environment, CSE, provided by Open Simulation Platform. The framework developed in CSE allows for connected followers to be exchanged without changing the framework itself. An existing simulation model of Oden in the Simulator of Arctic Marine Structures, SAMS, with the ice-hull interactions should be connected to a simulation model of the propulsion system onboard. Simulink and 20-sim were tested as software for the development of the model of the propulsion system. In conclusion, neither SAMS nor Simulink were compatible with CSE. Thus, all followers had to be developed in 20-sim. Since SAMS could not be coupled with CSE, ice loads had to be obtained from an independent simulation in SAMS of the icebreaker Oden running at a constant speed of 3 m/s through an ice field.

The co-simulation model contains errors and can therefore not be used to analyze the effect of ice loads on the propulsion line. The graphs from the co-simulation show that the simulation model of the engines and gearbox does not behave as expected, for example negative torque and rotational velocity are observed. Some of the error is probably due to an algebraic loop in the gearbox model, which there was not enough time to solve in a satisfying way. In the future, the co-simulation model of Oden could be developed in a better way as newer co-simulation software allows for the originally planned software for the followers to be used. The work on this thesis made it clear that this project has been conducted at a too early stage in the development process of the co-simulation software CSE. This resulted in a lot of time-consuming trial and error that could have been avoided if the project had been conducted after the final version of the co-simulation software was launched.

# Contents

<b>Preface</b>	<b>i</b>
<b>Acknowledgement</b>	<b>ii</b>
<b>Samandrag</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Abbreviations</b>	<b>vii</b>
<b>List of Symbols</b>	<b>viii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem statement . . . . .	1
1.2 Scope . . . . .	4
1.3 Limitations of the scope . . . . .	5
1.4 Structure of the thesis . . . . .	5
<b>2 Background information</b>	<b>6</b>
2.1 The arctic environment . . . . .	6
2.2 Ice capable ship . . . . .	8
2.3 Ice loads . . . . .	8
2.4 Digital twin . . . . .	9
<b>3 Theory</b>	<b>12</b>
3.1 Modeling a vessel . . . . .	12
3.1.1 Hydrodynamic resistance . . . . .	12
3.1.2 Engine . . . . .	12
3.1.3 Flexible coupling . . . . .	13
3.1.4 Gearbox . . . . .	13
3.1.5 Shaft . . . . .	13
3.1.6 Propeller . . . . .	13
3.1.7 PI-regulator . . . . .	14
3.2 Ice loads . . . . .	15
3.3 Bond Graph . . . . .	16
3.4 Co-simulation . . . . .	18
3.4.1 Follower . . . . .	19
3.4.2 FMU . . . . .	19
3.4.3 Signal communication . . . . .	20
3.4.4 Communication frequency . . . . .	21

3.5	Programs . . . . .	22
3.5.1	CSE - Core Simulation Environment . . . . .	22
3.5.2	MATLAB Simulink . . . . .	22
3.5.3	20-sim . . . . .	22
3.5.4	Simulator for Arctic Marine Structures . . . . .	22
<b>4</b>	<b>Co-simulation model of the icebreaker Oden</b>	<b>23</b>
4.1	Propulsion onboard Oden . . . . .	23
4.2	Ice load propagating into the propulsion system . . . . .	24
4.2.1	Hydrodynamic resistance . . . . .	25
4.2.2	Ice loads . . . . .	26
4.2.3	Engine . . . . .	26
4.2.4	PI-regulator . . . . .	27
4.2.5	Flexible coupling . . . . .	27
4.2.6	Gearbox . . . . .	27
4.2.7	Shaft . . . . .	27
4.2.8	Propeller . . . . .	28
4.3	Co-simulation model for the system . . . . .	29
<b>5</b>	<b>Results</b>	<b>32</b>
5.1	Input to co-simulation model . . . . .	32
5.2	Results from the co-simulation . . . . .	34
<b>6</b>	<b>Discussion</b>	<b>42</b>
6.1	Choice of DSB . . . . .	42
6.2	Simulation tools for the followers . . . . .	42
6.3	Monolithic or distributed simulation model . . . . .	43
6.4	Inclusion of the ice loads . . . . .	43
6.5	Consequence of assumption . . . . .	44
6.6	Uncertainties regarding the parameters used in the simulation . . . . .	44
6.7	The results from the co-simulation . . . . .	45
6.8	The new pre-release of the co-simulation software . . . . .	46
<b>7</b>	<b>Conclusion</b>	<b>47</b>
7.1	Recommendation for further work . . . . .	48
	<b>Bibliography</b>	<b>49</b>
	<b>Appendix I:Ice loads</b>	<b>a</b>
	<b>Appendix II: Systemstructure.ssd</b>	<b>b</b>
	<b>Appendix III: OspSystemstructure.xml</b>	<b>d</b>

## Abbreviations

ArcISo	Arctic Integrated Solutions AS
CPP	Controllable Pitch Propeller
CSE	Core Simulation Environment
DoF	Degree of Freedom
DSB	Distributed Simulation Bus
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
FPP	Fixed Pitch Propeller
IB	Icebreaker
NTNU	Norwegian University of Science and Technology
OSP	Open Simulation Platform
OSP-IS	Open Simulation Platform interface specification
PI	Proportional-Integrator
PID	Proportional-Integral-Derivative
rpm	Revolutions per minute
rps	Revolutions per second
SAMS	Simulator of Arctic Marine Structures
TCP/IP	Transmission Control Protocol/Internet Protocol
TVC	Torsional vibration calculation
ViProMa	Virtual Prototyping of Maritime Systems and Operation,
VPF	Virtual Prototyping Framework

## List of Symbols

$^{\circ}$	Degree
$^{\circ} C$	Degree Celsius
$\dot{v}$	Acceleration
$\eta_m$	Mechanical efficiency
$\nu$	Kinematic viscosity
$\omega$	Angular velocity
$\omega_{sp}$	Angular velocity of the set point
$\rho$	Density
$\tau_{env}$	Environmental loads
$\tau_{hydro}$	Hydrodynamic resistance
$\tau_{ib}$	Load from icebreaking
$\tau_{ice}$	Ice loads
$\tau_{ir}$	Load from ice rubble
$\tau_i$	Total load
$\tau_{prop}$	Ship propulsion
$\tau_x$	Load in x-direction
$\tau_y$	Load in y-direction
$A_f$	Foil area rudder
$A_w$	Wet surface area
$C_f$	Total hydrodynamic resistance coefficient
$C_{Dr}$	Drag coefficient
$C_{fp}$	Hydrodynamic resistance coefficient
$C_{fr}$	Hydrodynamic resistance coefficient
$D$	Diameter
$D_{RP}$	Drag on rudder port side

$D_{RS}$	Drag on rudder starboard
$D_R$	Drag on rudder
$e$	effort
<i>error</i>	error
$f$	flow
$h_c$	Height consolidated layer
$h_k$	Height keel
$h_s$	Height sail
$K_i$	Coefficient for integral gain
$K_p$	Coefficient for proportional gain
$K_T$	Thrust coefficient
$L_{coeff}$	Coefficient for modeling resistance
$L_{pp}$	Length between perpendiculars
$M$	Mass
$n$	Rotational velocity of propeller
$P$	Power
$p$	generalized momentum
$P_e$	Power engine
$q$	generalized displacement
$R$	Hydrodynamic resistance
$R_e$	Resistance engine
$R_n$	Reynolds number
$t$	Time
$T_i$	Torque engine
$T_{PP}$	Thrust propeller port side
$T_{PS}$	Thrust propeller starboard

$T_P$	Thrust propeller
$u_r$	Flow velocity across rudder
$v$	Velocity
$X$	Control signal



## List of Figures

1	Sea ice coverage on 18.09.19, and the red line is average coverage in periode from 1981-2010 . . . . .	1
2	Representation of what a digital twin can be . . . . .	3
3	Picture of the icebreaker Oden . . . . .	4
4	Oden propagating in Arctic pack ice . . . . .	7
5	Characteristics features of an ice ridge. . . . .	9
6	Hype Cycle for emerging technologies. . . . .	10
7	Failure modes that ice break in . . . . .	15
8	The nature of ice resistance as an average of longitudinal force . . . . .	16
9	Meaning of causal strokes. . . . .	17
10	Structure of the co-simulation . . . . .	19
11	FMU structure . . . . .	20
12	Signal exchange between FMUs . . . . .	21
13	Diagram showing the configuration of the propulsion system onboard IB Oden . . . . .	24
14	The propellers installed on Oden. . . . .	28
15	Structure for the co-simulation of one propulsion line with the intended followers and the software used for building the follower. . . . .	29
16	Structure of the monolithic simulation model built in 20-sim . . . . .	30
17	Simplified structure for the co-simulation . . . . .	31
18	Forces from breaking the ice . . . . .	32
19	Forces from ice rubble sliding and slamming along the hull of Oden . . . . .	33
20	Total ice load on the hull, which consist of the icebreaking and rubble force combined. . . . .	33
21	Load input to the propulsion system . . . . .	34
22	Total load on hull versus load on one propeller . . . . .	35
23	External load on one propeller . . . . .	35
24	Rotational velocity on the propeller. . . . .	36
25	Resistance on propeller . . . . .	36
26	Angular velocity over the gearbox . . . . .	37
27	Gearbox torque . . . . .	38
28	Gearbox torque, only the outlet towards propeller . . . . .	38
29	Torque delivered by the engines . . . . .	39
30	Plot shows the rotational velocity of the engines . . . . .	39
31	The difference between the rotational velocity of the engines and the controller set point . . . . .	40
32	Control signal from the PI-regulators . . . . .	41
33	Ice loads . . . . .	a

## List of Tables

1	Energy domains . . . . .	18
2	Technical data for IB Oden . . . . .	23
3	Parameters needed to calculate hydrodynamic forces on the hull of Oden . . . . .	26
4	Main parameters of ZA40S engine . . . . .	26
5	Characteristics for the flexible coupling . . . . .	27
6	Characteristics for the gearbox onboard Oden . . . . .	27
7	Main parameters for the propeller . . . . .	28
8	Co-simulation followers and contact points . . . . .	31

# 1 Introduction

## 1.1 Problem statement

On the top of the northern sphere of the Earth lies the Arctic, which consists of the Arctic Ocean covered by a gigantic ice sheet centered around the north pole. As a consequence of climate change, the temperatures are increasing, which causes the ice to slowly melt, reducing the size of the Arctic. Thus, making the Arctic more accessible. Nevertheless, the Arctic region is still an inhospitable area with extreme conditions and lacking of infrastructure. Therefore, emergencies have a higher risk of occurring, and also the consequences of these for humans, the environment, and equipment are far more severe. For a long time, the Arctic was cut off from human interaction, thus human interaction may cause damage to the pristine, fragile ecosystem.

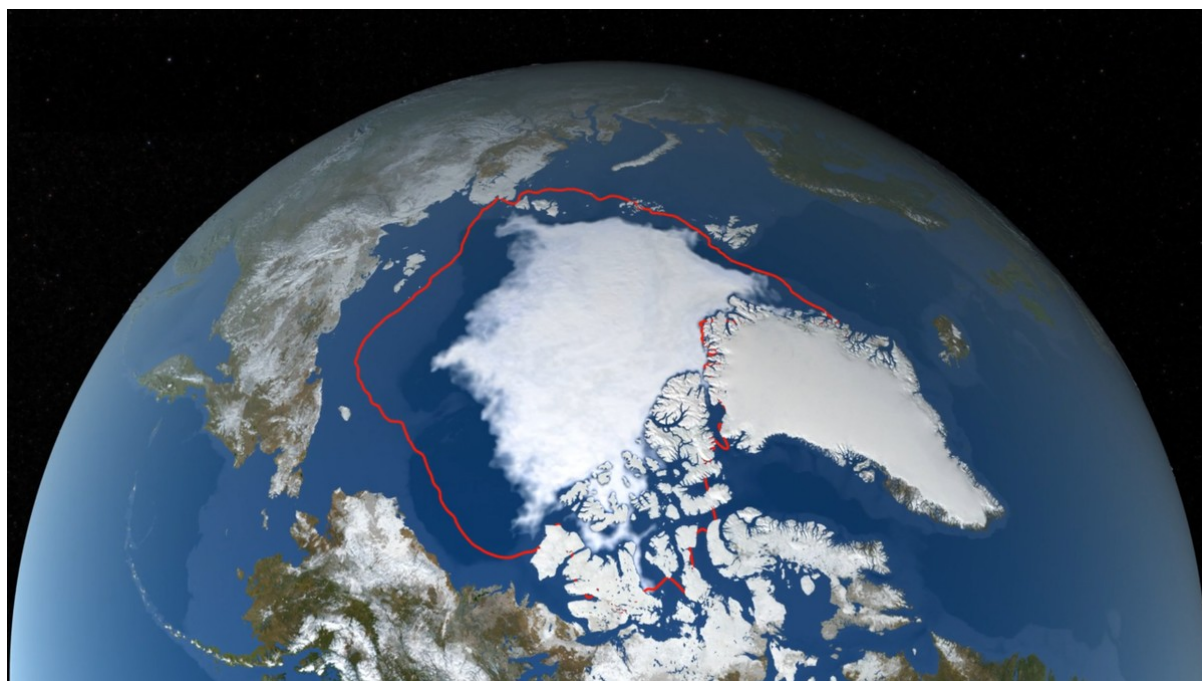


Figure 1: Sea ice coverage on 18.09.19, and the red line is average coverage in periode from 1981-2010. Photo: NASA's Scientific Visualization Studio; AMSR2 data courtesy of the Japan Aerospace Exploration Agency (JAXA); The Blue Marble data is courtesy of Reto Stockli (NASA/GSFC) (2019)

Therefore, when operating in the Arctic, specialized vessels are required to handle the additional environmental loads. There are three main types of vessels that operate in the Arctic and similar environments; ice-strengthened ships, ice going ships, and icebreakers. Out of these, icebreakers are the most capable of maneuvering through ice-covered water. They have strong hulls for crushing and breaking the ice. Additionally, a strong engine is necessary for the icebreaker to move forward in the ice. Furthermore, the propulsion system must be equipped to handle both loads of great magnitude and rapidly changing loads. Ice loads tend to build up as the ice push against the hull until the ice breaks and the load drops down before a second load build up as

the rubble slides and slams along the hull, this gives a zigzagging load pattern with great peaks and rapid drops. This zigzagging load pattern is hard to design for as the loads are extreme and varying.

Gaining more knowledge about how the zigzagging load pattern influences the hull, propulsion system and other systems onboard, would increase the safety of operations and the design of the vessels. Regarding the interactions between hull and ice, a simulation program exists which can simulate the loads on the hull. However, it is limited to only actions on the hull. For safer and more optimal designs, a better understanding of how this type of loading influences the structure as a whole and the mechanics onboard is needed.

The most common way of simulating a vessel is to simulate each system individually, which gives a non-optimal vessel, as each system is optimized individually to perform at its best. A vessel is a complex structure with many closely linked systems, they should be optimized for working together as one unit as opposed to separates. Through the usage of digital twin technology, these individual simulation models can be linked together to create a complete model of the entire vessel. By establishing the digital twin in the design phase, the vessel can be optimized as a whole prior to construction, thus avoiding unnecessary deficient design choices. The digital twin also allows for testing of the operating reliability, without constructing the physical vessel. By having a digital twin of an existing vessel, one will get a better overview of its health. For instance, using sensors to monitor wear and tear in combination with the digital twin, provides a better foundation for maintenance scheduling. Potential upgrades for software and components can be tested beforehand on the digital twin, giving valuable insight to the compatibility of the upgrade, and whether it is financially beneficial.

A benefit of establishing a digital twin is that changes can be tested and optimized before it is implemented on the real vessel. This may save time and money later on, as there will be fewer surprises. The digital twin also provides more knowledge of how to optimize the utility of the vessel in operations, as the operations can be simulated beforehand.

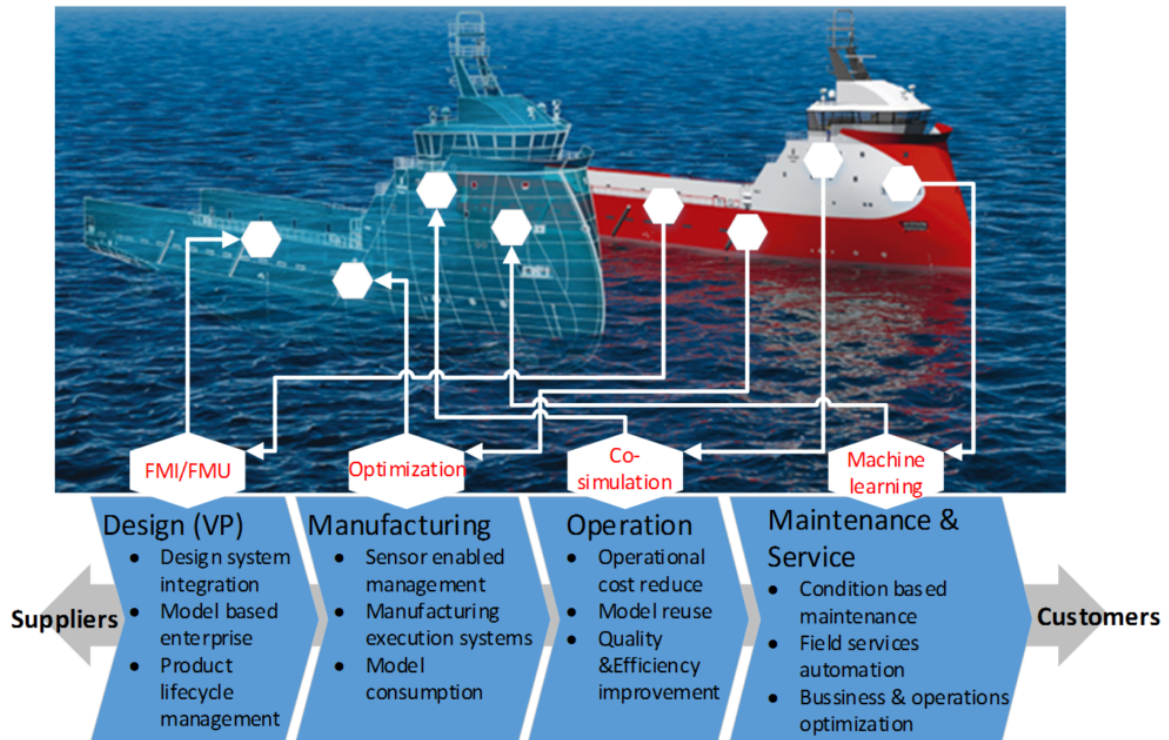


Figure 2: Representation of what a digital twin can be. Illustration from Intelligent Systems Lab @ NTNU in Ålesund (MechatronicsLab) (2020)

Thus the question remains, what is a digital twin? At its core, a digital twin is a digital representation of a physical asset. This can be as simple as an equation that tells something about the behavior of a real unit. It may also be a complex simulation model of a car, house, vessel, or even an environmental model. The digital twin can be built before the physical twin, which makes it a useful tool for developing and designing vessels.

The next question becomes, how to build a digital twin? This can be done in many ways depending on what it is a digital twin of. A complex construction, such as a vessel, requires an advanced simulation. One may build the full simulation model using only one simulation software. However, simulation software is often built to solve one specific problem, and a vessel consists of several parts that belong to very different engineering domains. The other option is co-simulation, which provides the opportunity to use specialized tools to simulate each system. Co-simulation has a modular approach to simulation and is very flexible, thus the simulation model can include hardware, software, and also humans in the simulation loop. This flexibility in the simulation allows for choosing the optimal simulation tool for each task and to have an incremental approach to building the model as one connected simulation is easily exchangeable.

Because vessels operating in ice are highly specialized to operate in the harsh conditions of the Arctic, it would be beneficial to investigate them using the digital twin technology as this can give insight to multiple systems and the interactions between them simultaneously. For a vessel to survive in the Arctic, each system has to be reliable and work optimally under rough

conditions, which is expensive and energy demanding. Therefore, the idea is that increased knowledge from analyzing the digital twin in form of a co-simulation model can provide more insight into how the ice loads impact the vessel and the equipment onboard it, not only limited to knowing the loads it has to overcome to be able to maneuver in the arctic environment. This would hopefully give vessels that can perform even better and be more reliable in the Arctic.

## 1.2 Scope

A suitable choice for testing and developing a digital twin would be to investigate an already existing vessel with a lot of accessible information, which operates in the arctic, and is already subject to research. The Swedish icebreaker Oden is a research vessel operating in both the Arctic and Antarctic, that fits the description. The icebreaker was built in 1988 at the request of the Swedish Maritime Administration. Originally Oden was constructed as an escort icebreaker, which means that she breaks the ice to provide safe passage for other vessels that needs help to manage the ice. Subsequently, Oden was equipped with a wide range of scientific equipment and has served as a research vessel on voyages to the Arctic and Antarctic. In 1991 she was the first non-nuclear icebreaker to reach the North-Pole.



Figure 3: Picture of the icebreaker Oden. Photo: Larsson & United States Navy (2008)

This thesis focuses on co-simulation, and explores the possibilities of the co-simulation software Core Simulation Environment, CSE. In order to do so, the icebreaker Oden is chosen as subject for the simulation, and in particular to investigate how the ice loads act on the hull and the influences on the propulsion line as the icebreaker propagates through the ice field.

CSE is a co-simulation tool that connects different simulation models. To build the simulation model of Oden, the different system is created using different simulation software. The FMI-standard is used to make FMUs of the different simulation models that are connected in CSE. The propulsion system should be connected to the simulation model in Simulator for Arctic Marine Structures, SAMS, that simulates the loads from the ice-hull interactions for Oden.

### **1.3 Limitations of the scope**

The scope of this thesis does not include making a simulation model of the ice loads acting on the hull of Oden, only to use the existing simulation model of the icebreaker Oden in SAMS.

Further, the scope of the thesis is to make the co-simulation model using CSE, other software could be used like Coral, that was used in the project thesis, or Simulink, but this is considered outside the scope.

Using other co-simulation software to compare and validate the results obtained using CSE is also considered outside the scope of the thesis. This is considered as a task for future work.

### **1.4 Structure of the thesis**

Chapter 1 gives an introduction to the thesis and problem statement.

Chapter 2 gives an introduction to the arctic, vessels in ice, ice loads and state of the art when it comes to digital twin and co-simulation.

Chapter 3 digs deeper into the theory behind the co-simulation model.

Chapter 4 describes co-simulation model. How each part is modeled and the structure of the co-simulation.

Chapter 5 present the results from the co-simulation.

Chapter 6 discuss and reflect upon the chosen programs, models and results previously presented.

Chapter 7 gives a conclusion to this thesis and suggestions for future work that should be done on this field.

## 2 Background information

The Arctic is located on the northern hemisphere, but exactly what the Arctic is is not completely agreed upon. There are three main definitions that are commonly used for where the Arctic exactly is. The first being the Arctic Circle, which is located at  $66^{\circ} 33'N$ , but this definition includes areas with a non-typical Arctic climate and exclude other areas with an Arctic climate, like Greenland. The second definition is the tree limit, which is one of the worlds major ecological boundaries (Smithson et al. 2002). The tree limit is an ecological boarder were trees are not able to grow north of. Occasionally there are some trees north of the tree limit, as this is a gradual boarder where the landscape changes from forest to tundra. The third definition is the  $10^{\circ}$  isotherm, which defines the Arctic as the area where the average temperature in the warmest month is below  $10^{\circ}C$  (Smithson et al. 2002). The  $10^{\circ}$  isotherm is roughly 100-200 km north of the three limit.

### 2.1 The arctic environment

Arctic areas consist of both oceans and land area, where the ground is frozen with permafrost with as an active layer on top. The temperatures are low, averaging at  $10^{\circ}$  iin the warmest months In spite of the absence of trees, some vegetation manages to grow in the Arctic, examples of which are some grasses, sedges, shrubs, mosses, lichens, and flowers (Smithson et al. 2002). During the Arctic summer, there is midnight sun, and in the winter month, polar night, which means absolutely no sun. Like most other areas there are differences between coastal climate and inland climate, typically the inland is colder and drier than the coastal areas. Most areas get less than 250 mm precipitation annually, but exact precipitation is difficult to measure since about 60% of it is snow (Smithson et al. 2002).

Large parts of the Arctic basin is covered by ice, this icecap grows during the winter months and melts during the summer. With global warming, the temperatures in the Arctic are increasing, which can be seen in the shrinking of the icecap and the tree limit moving further north. Thus the Arctic itself is shrinking, given the  $10^{\circ}$  isotherm and tree limit definition.





Figure 4: Oden propagating in Arctic pack ice. Photo: Linder (2008)

The Marine arctic environment is characterized by the low temperatures, icy water, rapidly changing weather, cloudiness, and remoteness of this region. The low temperatures of the Arctic Ocean are typically in the range of approximately  $-25^{\circ}\text{C}$  in winter to  $0^{\circ}$  in summertime (Walsh 2008). The precipitation is low over the ocean, and the area is very cloudy (Walsh 2008). The Arctic Ocean is covered by floating sea ice and occasionally some icebergs and ice ridges, the sizes vary from small pieces to massive ice sheets.

Besides, the Arctic environment can be windy, in combination with the low temperatures marine icing can become a major challenge (Kulyakhtin 2018). The marine spray icing occurs as the vessel wave interactions create sea-spray droplets that freeze onto the vessel. The consequences of this can be instability as the weight of ice on top of the vessel becomes so heavy that it changes the metacentric height. Additionally, the marine spray icing can result in a slippery deck, and that exits, ventilation, and emergency equipment gets covered in ice and can not be used.

## 2.2 Ice capable ship

The Arctic is an area of economic interest, and the region is important for Arctic fisheries, cargo transport through the northwest and northeast passage, tourism, science, and also oil and gas.

Vessels designed for ice infested environments have thicker hulls, sea chests must be protected from ice, and often both propeller and rudder are reinforced for higher loads than ships normally experience when operating in open-water. Vital functions such as fuel tanks and ballast tanks may be heated to ensure that vessel can operate properly. Which special features the ship has is depending on the purpose of the vessel and the class assigned. Three types of vessels operate in the Arctic (Heyn 2019):

- Ice strengthen ships
- Ice going ship
- Icebreakers

Ice strengthen ships are mainly designed for open-water, but have reinforced hulls thus they are capable of withstanding the ice loads. To operate in the ice, these vessels are dependent on the ice observation system, which are mostly expert onboard and ice predictions form land-based forecasters. When the ice conditions get heavy, ice strengthen ships are dependent on support from icebreakers to manage the ice. These vessels are typically cargo, cruise, or drilling vessels.

Ice going ships are designed to be independent of help from icebreakers, but occasionally they get locked in the ice and need icebreaker support. They are designed with reinforced hulls and have stronger engines compared to their open-water counterparts. Typical tasks for an ice going vessel is to be a supply vessel for Arctic offshore operations.

Icebreakers are the most capable of the ship types that operate in an Arctic environment. Their only purpose is to break the ice, thus they have reinforced hull to withstand all ice conditions. The engines are huge to be capable of breaking ice, which can be several meters thick. Propellers are strengthened to be able to mill ice if it gets stuck there. Icebreakers are used for scientific purposes and as part of the ice management for other ships, the typical task can be escorting a vessel or break a channel in the ice so a drilling vessel can perform drilling operation.

## 2.3 Ice loads

When the temperature of seawater decreases below the freezing point, sea ice can be formed. As sea ice is an inhomogeneous material where the material properties like density, porosity, salinity, and thickness are constant for a given region (Heyn 2019), but changes with the temperature and age of the ice. Second-year ice and multi-year ice, which has survived one and at least two summers respectively, is typically stronger than first-year ice (Løset et al. 2006). The increased strength and reduced porosity is the result of more time to grow, reorientate, and transporting out the brine. Brine is saline water trapped inside the ice.

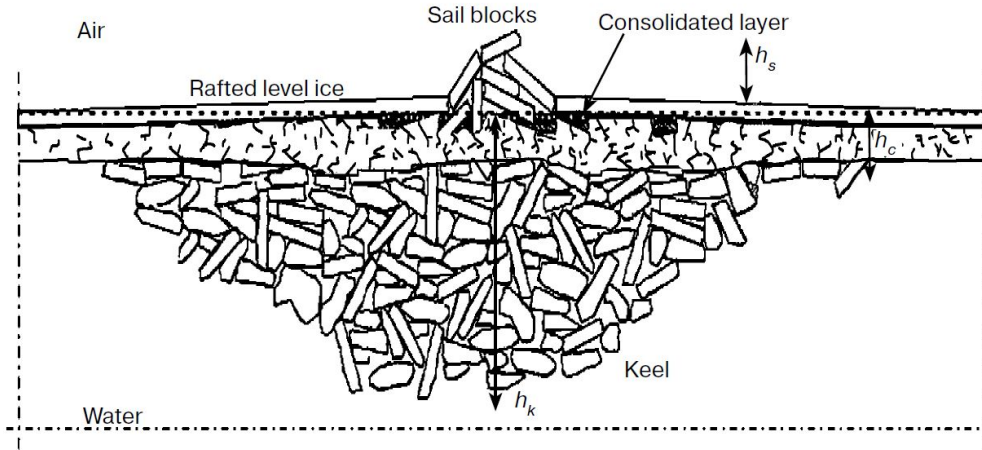


Figure 5: Characteristics features of an ice ridge. Illustration from Løset et al. (2006)

Undisturbed sea ice that freezes into an even layer is called level ice. Rubble ice is broken ice, which includes both floating and grounded ice blocks. Ice ridges are shaped from compressed rubble ice and form a pile of broken ice. An ice ridge, which can be seen on Figure 5, has a keel, sail, and consolidated layer. The keel is the submerged part, and the sail is the part above water level, and the consolidated layer is the part of the ridge that is refrozen (Løset et al. 2006). Additionally to sea ice, a vessel could encounter icebergs, which are freshwater ice orienting from land and calved into the ocean.

Sea ice can be hazardous to vessels if the magnitude of the loads from the ice hull interactions surpasses the design load of the hull. An ice load occurs as a pressure on an area, and is defined as a local load if it acts on a restricted area. This is different from e.g. wave loads which can be defined as a global load as it acts on the entire ship. The ice load varies depending on several factors such as the velocity of the vessel, hull design, drifting direction of the sea ice, and the structural properties of the sea ice, which is an inhomogeneous material. A well-designed hull for maneuvering in ice will be capable of reducing the ice load by redirecting them, lifting the vessel, crushing and breaking the ice.

## 2.4 Digital twin

A digital twin is a digital representation of a real asset. This can be either a digital drawing or a mathematical equation, which is also a digital representation, or a sophisticated dynamic simulation model which has real-time contact with the physical twin, or anything in between. The level of intricacy does not affect if the digital representation of an item has status as a digital twin. However, it is common to assume that the digital twin is highly advanced. The digital twin can be used to develop and test new designs, when planning and building the ship, to plan operations and maintenance, and testing future upgrades and modifications (Ludvigsen 2019). One digital twin may be used in three phases, first being the design phase, the second construction phase, and the third being the operational phase.

By having sensors installed on the physical twin and collecting data, these can be used for building a more accurate version of the digital twin. Additionally, the digital twin can be synchronized with the physical twin in real-time, this can be useful when planning operations and maintenance. Based on the data collected from the physical vessel and digital twin, the vessel performance and utilization can be analyzed in order to optimize the fuel consumption and energy flow, and monitor component condition for more optimal maintenance (Stensvold 2016).

## Hype Cycle for Emerging Technologies, 2018

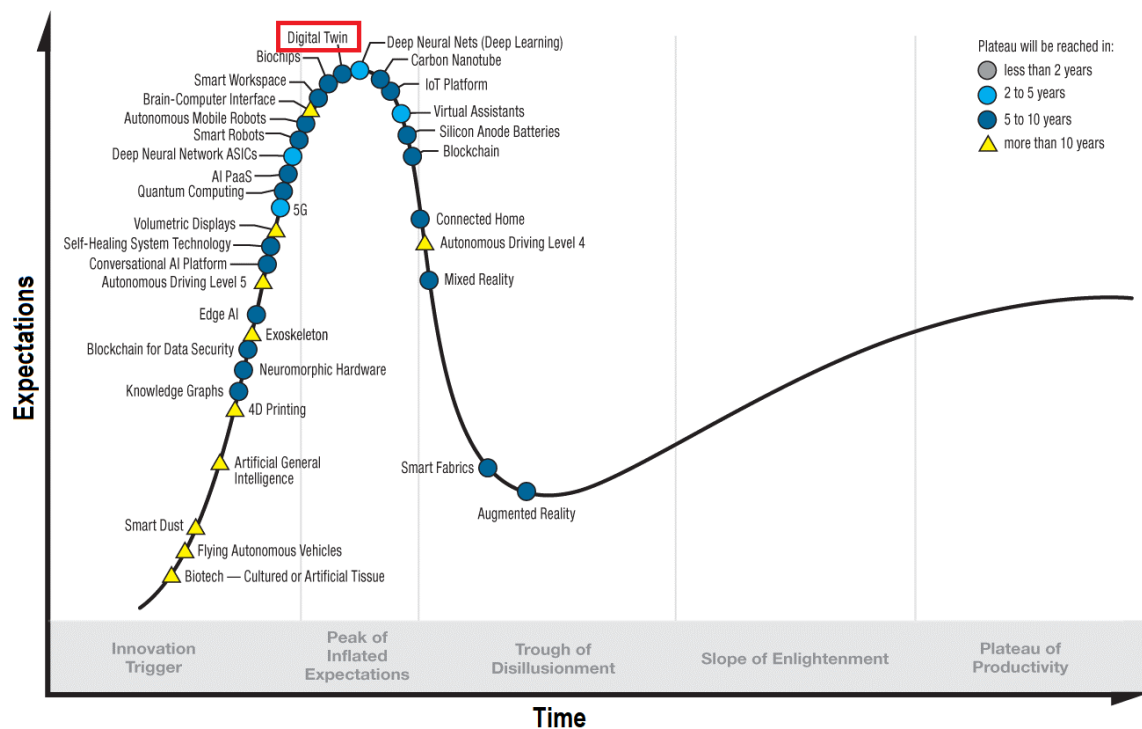


Figure 6: Hype cycle for emerging technologies. Illustration from Gartner, Inc. (2018)

In the year 2018, the digital twin technology was at the top of the Gartner Hype Cycle for emerging technologies (Gartner, Inc. 2018). This curve provides an annual overview of technological trends in the market. When one technology is at the peak of the curve, the expectations are high and the technology is in the spectra of "Peak of inflated expectations". This is defined as the phase where there is "much talk" and several publications addressing the successes and failures of the technology. The technology has attracted the interest of many actors in the market, but there are only a few of them that take action at this stage in the development of the digital twin technology.

The following year, the digital twin technology was not explicitly mentioned on the top 29 list in the 2019 report. There are two reasons why the digital twin was not mentioned that year, one reason being that it has been on the list for some years already. The second reason is that it has become too integrated into the businesses, both reasons make the digital twin a non-emerging technology. Further Gartner states that there are five emerging areas of technology today, one of them is Digitalized ecosystems. The digital twin technology is an important part of the technology that enables digital ecosystems and builds bridges between humans and technology.

There exist several different types of digital twins and co-simulation software in different business segments today, however, most of them are company-specific, in-house software, which due to privacy reasons, are not widely spread amongst users. In the maritime industry, there are some business initiative towards making digital twin an available solution for developing better, more reliable, and economic designs for vessels as well as making the operational phase safer and more cost-effective. Some of them are described below.

The Virtual Prototyping of Maritime Systems and Operation, ViProMa, was established back in 2013 and continued until 2016. The intention behind the project was to create an open-source virtual prototyping framework, VPF, that was an easy co-simulation tool to collaborate in for the industry. The tool for co-simulations needed to be able to connect simulations models from different engineering disciplines and also protect business secrets. This would make the maritime industry ready face the new challenges and make the maritime industry safer and more efficient. The ViProMa project enables designers, engineers, shipbuilders, and owners to test and evaluate the performance of the concept in terms of efficiency, performance, and cost before a vessel is built, even before the building process is started (Kyllingstad 2018). The resulting VPF program was named Coral.

The *Open Simulation Platform*, OSP, is a joint industry project where initiated by NTNU, DNV-GL, Sintef, and Kongsberg in 2017. The intention behind OSP is to enable the construction of digital twins. OSP currently consists of time-domain models of components, virtualized communication network, software-driven control algorithms. The vision behind OSP is to create an ecosystem for collaborative digital twin simulations that can solve challenges related to design, commissioning, operating, and assuring complex, integrated operating and assuring complex, integrated systems (Kyllingstad 2019, Ludvigsen 2019).

## 3 Theory

### 3.1 Modeling a vessel

This chapter presents the theoretical background for modeling a vessel.

#### 3.1.1 Hydrodynamic resistance

The hydrodynamic force acting on the hull is dependent on the velocity of the vessel. The linear surge resistance of the hull consists of viscous effect and hull roughness. The hydrodynamic resistance is calculated in one degree of freedom using Equation 1.

$$R = \frac{1}{2} C_f \rho A_w v^2 \quad (1)$$

Where  $R$  is the total resistance,  $C_f$  is the total resistance coefficient, which is dimensionless.  $\rho$  is the density of seawater, which is  $1025 \text{ kgm}^3$  at  $25^\circ\text{C}$ .  $A_w$  is the wet surface area of the vessel. The coefficient  $C_f$  can be split into two terms, as seen in Equation 2.

$$C_f(v) = C_{fp}(v) + C_{fr} \quad (2)$$

Both  $C_{fp}$  and  $C_{fr}$  are coefficients for calculating the hydrodynamic resistance. The coefficient  $C_{fr}$  in Equation 2 is a constant and can be found from open water trials, as by Kramers (2016). In the first term,  $C_{fp}$  is a function of Reynolds numbers, which again is a function of velocity.

$$C_{fp}(v) = \frac{0.075}{(\log(R_n) - 2)^2} \quad (3)$$

$$R_n = \frac{v L_{pp}}{\nu} \quad (4)$$

Where  $R_n$  is Reynolds number and can be found using Equation 4.  $v$  is the velocity in  $\frac{m}{s}$ .  $L_{pp}$  is the length between perpendiculars in meters.  $\nu$  is the kinematic viscosity of seawater at  $0^\circ$  is  $1.787 \cdot 10^{-6} \text{ Pa} \cdot \text{s}$  (Kramers 2016).

#### 3.1.2 Engine

The torque of the engine is modeled as:

$$T_i = \frac{P_e}{\omega} \left( \frac{1}{\eta_m} \right) \quad (5)$$

Where  $T_i$  is the engine torque,  $P_e$  is the engine power,  $\omega$  is the angular velocity of the engine and  $\eta_m$  is the mechanical efficiency.

The resistance of the engine  $R_e$  is modeled as:

$$R_e = T_i \cdot (1 - \eta_m) \left( \frac{|v|}{\omega_{sp}} \right)^2 \quad (6)$$

Where  $T_i$  is the engine torque,  $v$  is the velocity of the engine,  $\omega_{sp}$  is the desired angular velocity of the engine and  $\eta_m$  is the mechanical efficiency.

### 3.1.3 Flexible coupling

Flexible couplings are useful to have between engines and gearboxes, or other rotational objects, as they can transfer torque from one rotating unit to another. Flexible coupling can make up for small misalignments and damps vibrations, shock loads, and noise. The flexible couplings are modeled using a bond graph consisting of a capacitance- and resistance-element.

### 3.1.4 Gearbox

The gearbox is configured with the engines on one side, and a shaft towards the propeller on the other side. The purpose of the gearbox is to make the engines work together and adjust the rotational velocity from the engines to a suitable size for the propeller. A gearbox can, in its simplest form, be described as cogwheels, where the different sizes of the cogwheel give the required change in angular velocity.

The gearbox is model using inertia elements with a transformer element in between. The transformer element models the gear ratio, while the inertia elements models the rotating inertia of the cogwheels.

### 3.1.5 Shaft

The shaft in a propulsion system is a metal rod that transfer rotational velocity and torque from one unit to another in the propulsion line. A shaft has both rotational inertia and rotational stiffness, which is modeled using the bond graph elements inertia and capacitance.

### 3.1.6 Propeller

There are several different types of propellers. Fixed pitch propeller, FPP, as the name suggests, has fixed pitch on the blades of the propeller, whereas a controllable pitch propeller, CPP, can change the pitch of the propeller. Having a CPP is useful on an icebreaker, as it is possible to change the amount and direction of the produced thrust by changing the angle of the propeller blades.

The relationship between the thrust produced and the propulsion of the vessel is given as:

$$\tau_{prop} = T_{PP} + T_{PS} - D_{RP} - D_{RS} \quad (7)$$

Where  $\tau_{prop}$  is the propulsion of the vessel,  $T_{PP}$  and  $T_{PS}$  is the trust produced by the propeller on port side and starboard respectively. The thrust produced by one propeller  $T_P$  is given by Equation 8:

$$T_p = \rho D^4 K_T |n|n \quad (8)$$

Where  $\rho$  is the density of seawater,  $D$  is the propeller diameter and  $n$  is the rotational velocity of the propeller.

$D_{RP}$  and  $D_{RS}$  in Equation 7 is the drag force on the rudder on port side and starboard respectively. The drag force on the rudder  $D_R$  is the drag force on the rudder given in Equation 9.

$$D_R = \frac{1}{2} \rho A_f C_{Dr} u_r^2 \quad (9)$$

Where  $\rho$  is the density of seawater,  $A_f$  is the foil area of the rudder,  $C_{Dr}$  is the drag coefficient and  $u_r$  is the flow velocity across rudder.

### 3.1.7 PI-regulator

The control system for the engines is modeled as a proportional-integral regulator, PI-regulator.

$$X = K_p \cdot error + K_i \int error dt \quad (10)$$

Where  $X$  is the control signal,  $K_p$  is a coefficient for the proportional gain,  $K_i$  is the coefficient for the integral gain,  $error$  is the difference between the engine setpoint and the delivered thrust.



### 3.2 Ice loads

For a vessel to move forward in ice-covered water, it has to break through the ice. The magnitude of the ice load also depends on which failure mode the ice breaks in (Løset et al. 2006). There are six classified failure modes, they are creeping, radial cracking, buckling, circumferential cracking, spalling, and crushing, illustrated in Figure 7 (Sanderson 1988). Whereas the last one, crushing is the most intensive as the ice gets pulverized and there is no pressure relief from the ice. Cracking, both radial and circumferential, gives pressure relief, and thus lower ice loads.

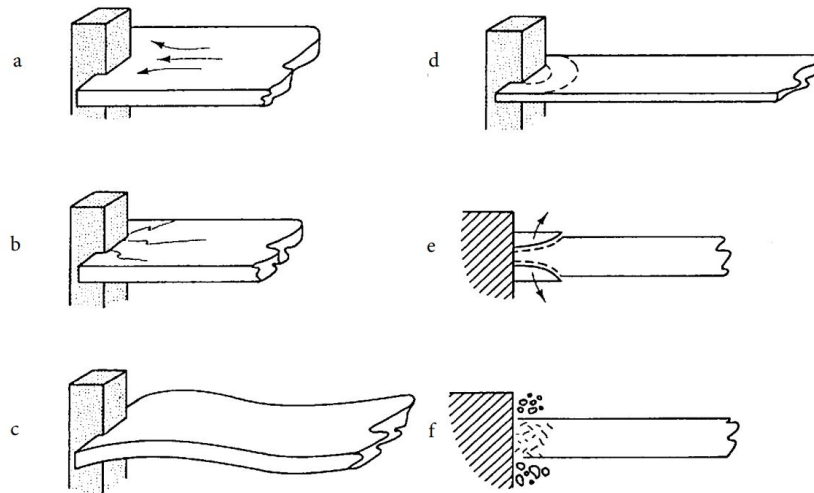


Figure 7: Failure modes that ice break in. a) Creep, b) Radial Cracking, c) Buckling, d) Circumferential cracking, e) Spalling, f) Crushing. Illustration from Sanderson (1988)

Breaking the ice can qualitatively be described as a four-step breaking process. The ice load increases in the first phase, as the first crushing, deflecting, and bending occurs. The second phase starts when the ice fails because the pressure is at the maximum which the ice can handle. This results in ice failure, and a load drop as the ice breaks rotates and gets pushed away. When the ice slides underneath the hull, slamming and ventilation can occur. In the third phase, there is another load build-up due to friction loads caused by the ice sliding along the hull. Lastly, in the fourth phase, the ice slides off the hull or get milled in the propellers. The four phases occur simultaneously in several places along the hull (Heyn 2019).

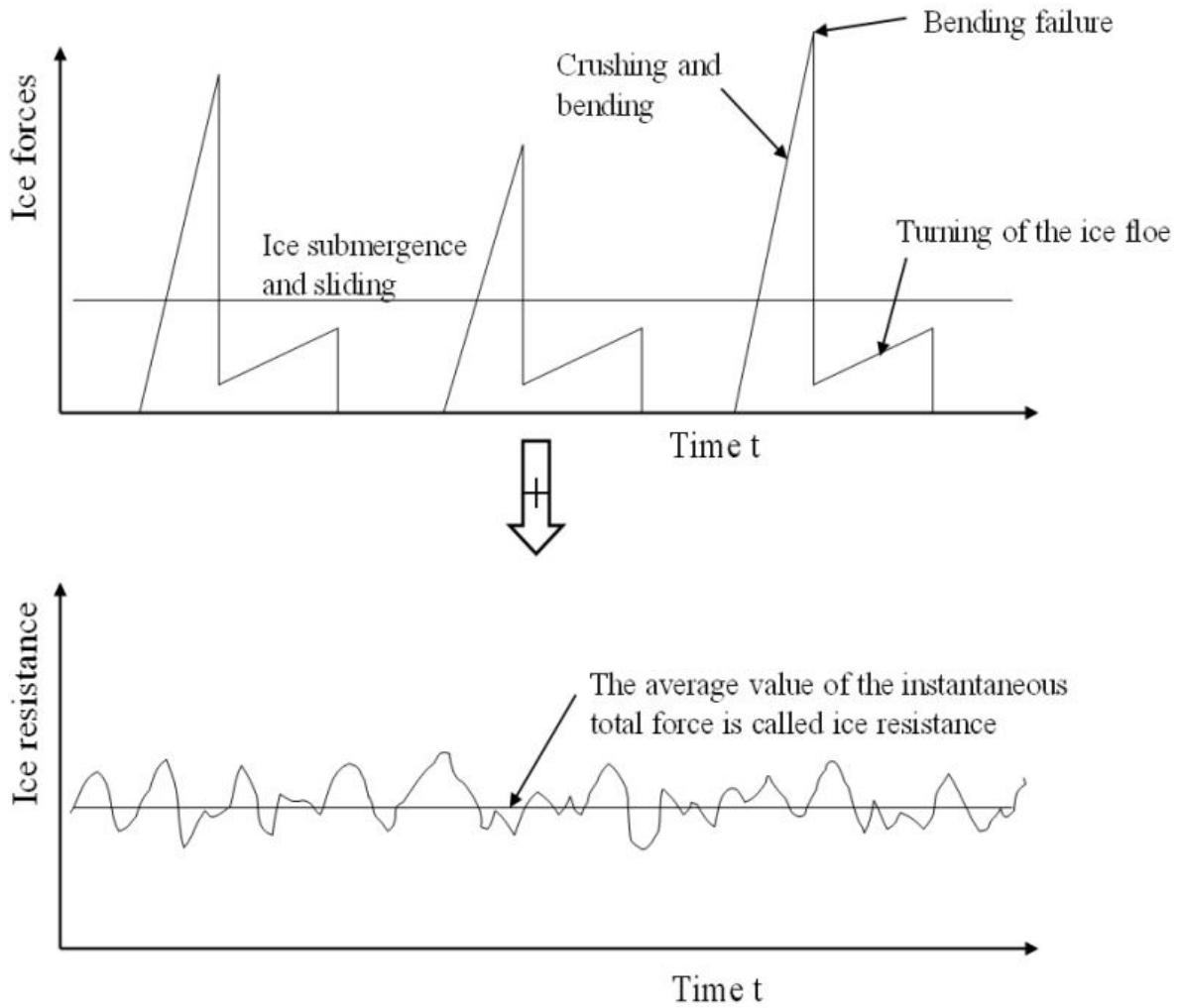


Figure 8: The nature of ice resistance as an average of longitudinal force. Illustration from Riska (2011)

### 3.3 Bond Graph

The principles behind the bond graph method, and the dynamics of leader-follower interactions have much in common, therefore the bond graph method is convenient when building dynamic followers. The bond graph method is based on the principle that all dynamic systems can be modeled in terms of energy transfer between elements (Pedersen & Engja 2014). It is a method for developing mathematical models of the system dynamics and energy transfer. Because energy is defined the same way in all physical systems, models can be built consisting of elements from multiple domains (Pedersen & Engja 2014).

$$P = e \cdot f \quad (11)$$

Equation 11 shows that power is a function of effort,  $e$ , and flow,  $f$ . Effort and flow are *power*

variables. There are two additional power variables,  $p$ , generalized momentum and  $q$ , generalized displacement, which are the integrals of effort and flow respectively.

$$E(t) = \int_0^t P(t)dt = \int_0^t e \cdot f dt \quad (12)$$

Equation 12 shows the relationship between energy and power.

The elements in a bond graph uses power bonds to transfer signals. The power bonds are represented as half arrow assigns with an arrow pointing in the positive power direction (Pedersen & Engja 2014). The bar on the arrow is a causal stroke, which can be seen on Figure 9 where A and B are representation of simulation model or a bond graph element. A causal stroke next to the element indicates that the element has effort as input and flow as output. The causal stroke and power bond direction are independent information.

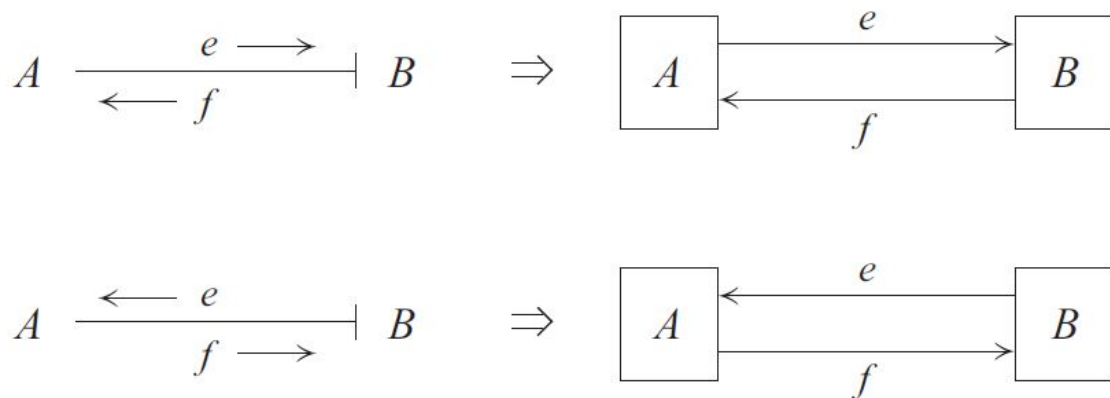


Figure 9: Meaning of causal strokes. Illustration from: Pedersen & Engja (2014)

The advantage of using the bond graph approach is that it simplifies the simulation of multi-domain systems. Table 1 provides an overview of the different energy domains that can be represented by a bond graph. When simulating a diesel-electric-propulsion system one needs to be able to simulate mechanical rotation and electrical parts that are connected, for which a bond graph is suitable.

<b>Energy domain</b>	<b>Effort</b>	<b>Flow</b>	<b>Momentum</b>	<b>Displacement</b>
<b>Mechanical translation</b>	Force	Velocity	Linear momentum	Distance
<b>Mechanical rotation</b>	Angular velocity	Torque	Angular momentum	Angle
<b>Hydraulic</b>	Pressure	Volume flow rate	Pressure momentum	Volume
<b>Electrical</b>	Voltage	Current	Flux linkage	Charge
<b>Thermal</b>	Temperature	Entropy flow	(not defined)	Entropy
<b>Magnetic</b>	Magnetomotive force	Flux rate	(not defined)	Flux
<b>Chemical</b>	Chemical potential	Rate of reaction	(not defined)	Advancement of reaction

Table 1: Energy domains (Pedersen & Engja 2014)

### 3.4 Co-simulation

Co-simulation is a method for building simulation models that consist of several simulation models joined together. This makes it possible to have simulation models with sub-simulators distributed on different computers and software and to include hardware, such as sensors, and human interaction (Skjong 2017). The flexibility in where and how the simulation models are made allows making and combining specialized simulation models to make more accurate representations of the physical asset. Co-simulation is also making it easier to collaborate on building complex simulation models, as one can build several models and combine them, as opposed to one massive simulation model.

By having the simulation models distributed over several independent software, the need for a method for connecting the sub-simulators arise. This can be solved by using a distributed simulation bus, DSB, which acts as an interface for the co-simulation and ties the distributed systems together (Rindarøy et al. 2015). The DSB is the framework for the simulation. This framework is set up as a leader-follower framework, or commonly addressed in the literature as the master-slave, but in the light of recent events and history such terminology should be avoided (Shankland 2020). A leader-follower structure for a co-simulation can be seen in Figure 10, where the co-simulator on the top is the leader, and the connected sub-simulators are the followers. The purpose of having a leader for simulation is to organize the communication between the followers, so everyone knows what to do and when to do it (Rindarøy et al. 2015).

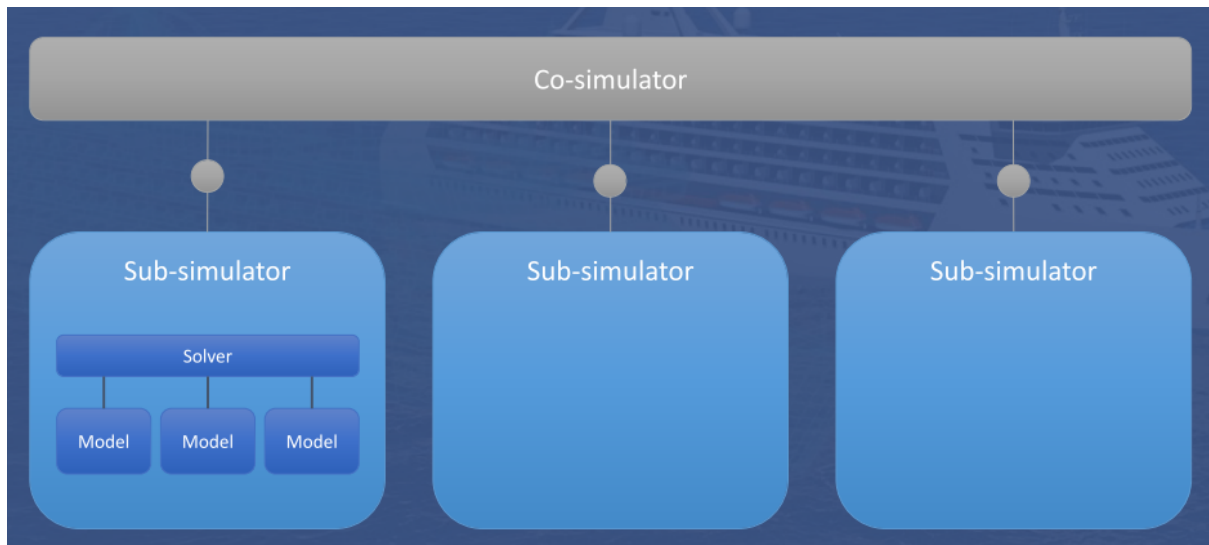


Figure 10: Structure of the co-simulation. Illustration from Kyllingstad (2019)

On Figure 10 the follower is named sub-simulator. A follower must contain a solver that works independent of the leader, and also one or more models. In order for the co-simulation to work, there needs to be a connecting interface. One is the FMI-standard, which is further explained in subsection 3.4.2. Due to the standardized interface of the followers, they can be exchanged without editing the co-simulation framework.

### 3.4.1 Follower

The connected simulators are often referred to as followers or FMUs, as they are exported as FMUs. The followers contain simulation models, which combined with the correct configuration, give the total simulation model for the entire system. Followers for co-simulation have to be designed in such a manner that they can exchange signals, which can be solved between the time steps. A follower contains individual models with both a mathematical description of the system and solver. There could also be several simulation models within the same follower.

Even though all followers are technically identical, it can be useful to divide the followers into two groups, dynamic followers and control followers (Rindarøy et al. 2015). The group called dynamic followers contains followers built on physical laws, which communicate through energy signals and can take control signals as input. Control followers are models of control systems which use non-energy signals as input and output and can take energy signals as input to monitor the state of the systems in the dynamic follower.

### 3.4.2 FMU

In order to have a standardized interface for the connection between the leader and follower, the Functional Mock-Up Interface, FMI, the standard was chosen by the developers in the OSP and became the relevant standard to discuss in this setting. The FMI-standard defines both the container and interface for a black-box exchange of dynamic files. In the co-simulation

context, it is used to ensure that the followers are exported as Functional Mock-Up Units, FMU. This makes modifications such as changing input, output, parameters, or model equation difficult. However, black-box implementation enables sharing models that contain sensitive data and business secrets without being spoiled. To make changes in the FMU, the model itself has to be adjusted, and the follower exported again as an FMU (Rindarøy et al. 2015).

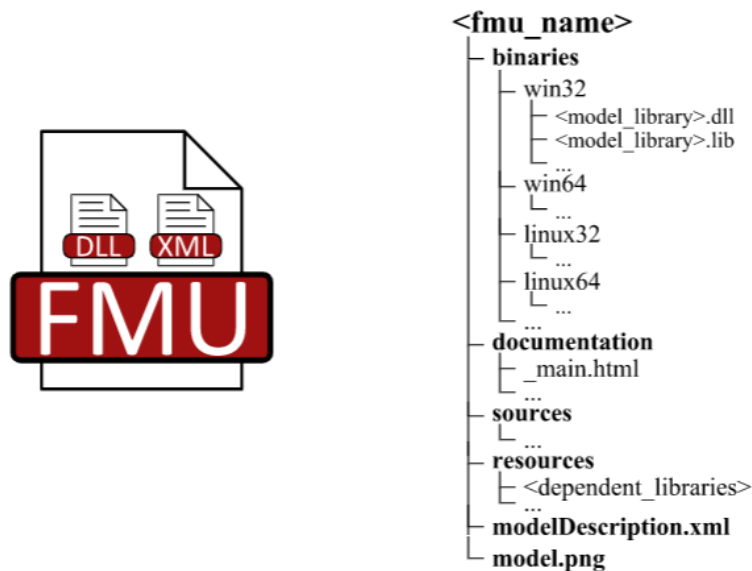


Figure 11: FMU structure. Illustration from Skjong (2017)

In the FMI-standard there are two types of FMUs, FMU for Model exchange and FMU for Co-Simulation, where the latter is the one used in co-simulation (Blochwitz et al. 2011). When using a Windows operating system, as in this thesis, the FMU would consist of shared libraries, dll-file, and xml-file containing metadata and supporting files (Skjong 2017).

### 3.4.3 Signal communication

The most important task for the leader is to organize the communication between the followers. It has to receive information and pass it on to the correct follower to ensure there is a dynamic relationship between the followers. Figure 12 illustrates the signal exchange between the FMUs. To ensure that there is a cause-effect relationship between the models, the signal needs to flow in both directions (Rindarøy et al. 2015).

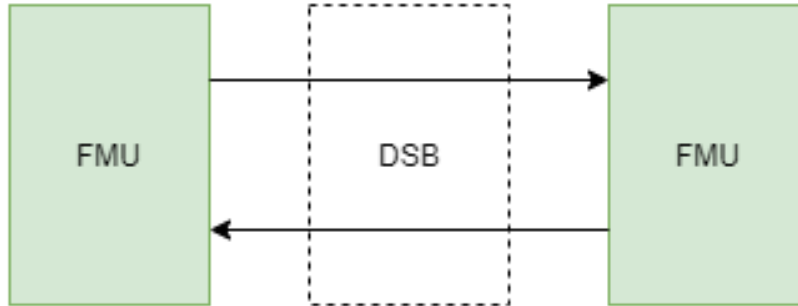


Figure 12: Signal exchange between FMUs

#### 3.4.4 Communication frequency

Both discrete and continuous systems can utilize the modality that the co-simulation offers to solve coupled problems. The exchange of data between followers is limited to discrete communication point only. Between each communication points, the subsystems are solved independently, as each follower has its own solver, making the data exchange work as synchronization of the followers (Blochwitz et al. 2011). The leader synchronizes the follower at data exchange frequency referred to as macro frequency, whereas the follower solves the internal simulation model at a micro frequency (Rindarøy et al. 2015). The micro frequency has to be higher than the macro frequency.

### 3.5 Programs

In order to make a co-simulation model, a digital twin has to be built. This requires specialized software. Simulation software suited for building the leader is software that can function as an administrator for the plugin simulation models. In this case, the type of connection used is FMU. A list of software that support FMU exports can be found at [fmi-standard.org/tools/](http://fmi-standard.org/tools/) (FMI-standard.org 2020). Both CSE and Simulink are potential hosts for a co-simulation model. While the other software presented in this thesis is intended for making followers, these need to be able to be exported as an FMU for the co-simulation to work. A simulation model in SAMS can not be exported as an FMU at this stage but is still relevant as it can simulate ice-hull interaction and is capable of being exported using a TCP/IP plugin.

#### 3.5.1 CSE - Core Simulation Environment

Core Simulation Environment, CSE, is an open-source co-simulation software provided by Open Simulation Platform. It works on both Windows and Linux and is a C/C++ library for integration of any application needed. It has a basic Graphical User Interface, to visualize the core capabilities and the results from running the co-simulation model (Open Simulation Platform 2019). The version of CSE that are used in this thesis is the CSE-v0.5.0-win64, which was pre-released in October 2019 (Open Simulation Platform 2020*b*).

#### 3.5.2 MATLAB Simulink

MATLAB Simulink is a graphical and text based programming environment, that are suited for modeling, simulation and analyzing multi-domain dynamical systems (MathWorks 2020). As Simulink has a wide range of built in libraries, features for realtime communications with sensors and creating digital twin it is a tool that could be used to create both the leader algorithm and the follower algorithm.

#### 3.5.3 20-sim

20-sim is a software for modeling and simulations of multi-domain dynamic systems (Controllab 2020). It is well suited for making dynamic followers for co-simulation purposes as the program supports export as co-simulation FMUs. 20-sim provided a wide range of modeling tools as one can use equations, block diagrams, physics blocks, and bond graph combined. The bond graph feature makes it easy to use in co-simulations, since energy transfer and causality are important concepts for both.

#### 3.5.4 Simulator for Arctic Marine Structures

Simulator for Arctic Marine Structures, SAMS, is a tool for investigating sea ice actions and the effects of sea ice on Arctic offshore structures (Tsarau et al. 2018). SAMS allows for modeling of marine structures and complex marine operations in the Arctic and subarctic (Tsarau et al. 2018). The structures in SAMS can be either floating or fixed. The structures, such as vessels or both fixed platforms, can also interact with a variety of environmental loads and ice conditions.



## 4 Co-simulation model of the icebreaker Oden

This chapter describes how the co-simulation model for Oden has been build. Firstly the main characteristics of the icebreaker is presented, before the machinery and the propulsion line is described in more detail as this is the most relevant for the co-simulation model. The model is firstly built in 20-sim as a monolithic simulation model. Thereafter the simulation model was split into fitting sizes for followers and exported as FMUs. Lastly the followers are reassembled using the co-simulation software CSE.

The Swedish icebreaker Oden is used for research purposes, and the main characteristics are given in Table 2.

<b>Oden</b>	
Length over all	107.75 m
Beam	31.20 m
Draft	8.50 m
Air draught	42.50 m
Displacement	13000 tonnes
Speed in open water	16 knots
Normal sea speed	11 knots
Icebreaking capacities	1.9 m level ice at 3 knots
Bollard pull	Continuous forward 250 tonnes
Main engines	4x8 cylinder Sulzer ZA 40S (each 4500 kW)
Main propellers aft	Two ducted controlled pitch propellers with one rudder each
Diesel generators	4x6 cylinder Sulzer AT 25 H (each 1200 kw)
Emergency generator	1x Cummins KTA-38G2 (600 kW)
Crew	23 persons, up to 50 scientists
Build year	1988
Building yard	Götaverken-Arendal AB (NB953)
Owner	Swedish Maritime Administration

Table 2: Technical data for IB Oden (Lubbad et al. 2018), (Swedish Maritime Administration 2019)

### 4.1 Propulsion onboard Oden

Onboard Oden, there are four marine diesel engines of the type Sulzer ZA 40S with 8 cylinders in line. Each engine produces 4500 kW, this gives a total of 18 MW produced power for propulsion. The propulsion system is a mechanical system consisting of the prime mover, gearbox, shaft, and propeller, which are split into two identical and separated propulsion lines. Power for use onboard the icebreaker is produced on separate diesel generators, there are installed four diesel generators of the type 6 cylinder Sulzer AT 25 H which can produce up to 1200 kW each. The configuration can be seen in Figure 13.

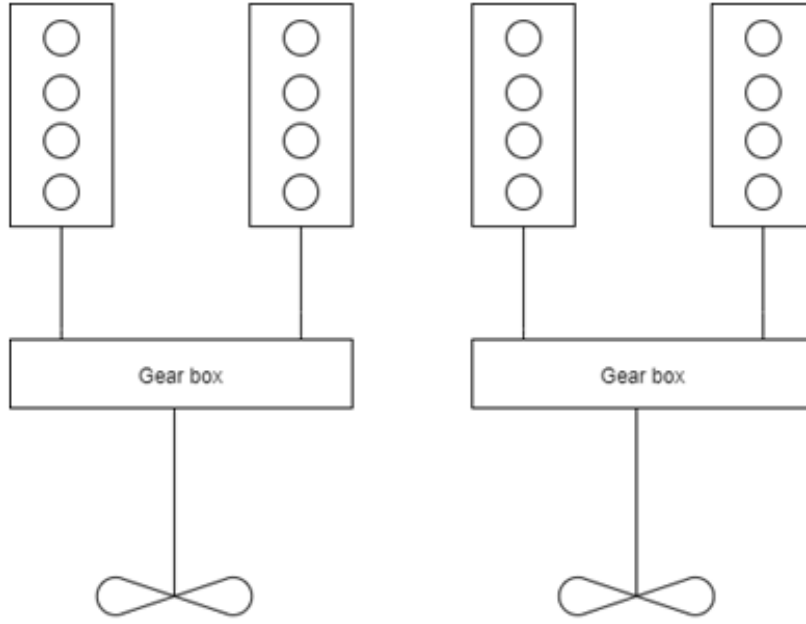


Figure 13: Diagram showing the configuration of the propulsion system onboard IB Oden

## 4.2 Ice load propagating into the propulsion system

The simulation model for the propulsion system consists of several submodels, which will be explained in more detail later on in this chapter.

The base for the simulation of the system starts out with Newton's second law. For the icebreaker it becomes as follows (Fossen 2011):

$$M\dot{v} = \tau_{prop} + \tau_{ice} + \tau_{hydro} + \tau_{env} \quad (13)$$

Where  $M$  is the mass of the vessel,  $\dot{v}$  is the acceleration,  $\tau_{prop}$  is the propulsion of the vessel,  $\tau_{ice}$  is the force from ice acting on the hull and  $\tau_{hydro}$  is the hydrodynamic resistance on the hull. The environmental loads,  $\tau_{env}$ , can be wind loads, icing, etc. that are acting on the vessel. For the simulation model, the environmental loads are not included in this stage of the simulation but can be added on a later stage. It is worth nothing that the ship propulsion is not the same as the thrust produced by the propeller. The relationship between the thrust produced and the propulsion is given in Equation 7.

As the ice load are recorded at a constant velocity, the simulation model also has constant velocity. The equation of motion/ newton second law for the system becomes

$$0 = \tau_{propulsion} + \tau_{ice} + \tau_{hydro} \quad (14)$$

Rewriting Equation 14 gives that the loads on the propeller is;

$$\tau_{propulsion} = -\tau_{ice} - \tau_{hydro} \quad (15)$$

Which is the basis for how the co-simulation framework is set up.

The list below gives a summary of the assumptions made for the simulation model.

1. The icebreaker propagates through the ice at a constant speed of 3 m/s. This indicates that the acceleration must be zero.
2. The external forces that act on the vessel are limited to hydrodynamic forces and ice loads, all other external forces are neglected.
3. Only one propulsion line is simulated as the loads are evenly distributed on both lines, thus the external loads are divided in half.
4. The engines are always on and are always producing positive torque.

#### 4.2.1 Hydrodynamic resistance

In order to find the hydrodynamic forces acting on the hull of Oden the equations presented in subsection 3.1.1 is used. Firstly one need to calculate Reynolds number using Equation 4 and insert it into Equation 2 to find the velocity dependent resistance coefficient  $C_{fp}$ . The non-velocity dependent coefficient  $C_{fr}$  is found from open water trials, and Kramers (2016) found it to be approximately 0.0066.

By inserting Equation 2 in Equation 1 on get that the total resistance on the hull can be found from Equation 16:

$$\tau_{hydro} = \frac{1}{2}(C_{fp} + C_{fr})\rho A_W v^2 \quad (16)$$

Table 3, below, has all the needed parameters needed to calculate the hydrodynamic force on Oden's hull at 3 m/s.

Wet surface	$A_w$	5000	$m^2$
Length between perpendiculars	$L_{pp}$	107.7	m
Velocity	$v$	3	m/s
Resistance coefficient	$C_{fr}$	0.0066	-
Resistance coefficient	$C_{fp}$	0.0019	-
Total resistance coefficient	$C_f$	0.0085	-
Density of seawater	$\rho$	1025	$kgm^3$
Kinematic viscosity	$\nu$	$1.787 \cdot 10^{-6}$	Pa s
Reynolds number	$R_n$	$180.8 \cdot 10^6$	-
Total resistance	$\tau_{hydro}$	196031	N

Table 3: Parameters needed to calculate hydrodynamic forces on the hull of Oden. (Kramers 2016)

#### 4.2.2 Ice loads

An ice load consists of both the icebreaking force and ice rubble force, which are added together. In the simulation model, a section of the simulation results is used as ice load input. As the icebreaker does not move in a straight line along with either of the x or y-axes, the total ice load is the sum of the load recorded in x and y-direction.  $\tau_{ice}$  consists of the icebreaking force,  $\tau_{ib}$  and ice rubble force  $\tau_{ir}$ .

$$\tau_i = \sqrt{\tau_x^2 + \tau_y^2} \quad (17)$$

$$\tau_{ice} = \tau_{ir} + \tau_{ib} \quad (18)$$

#### 4.2.3 Engine

The engine installed on Oden is a Sulzer 8ZAL40S, which has eight cylinders and a line configuration (Woodyard 2004). The characteristics of the engines are given in Table 4.

Bore	400	mm
Stroke	560	mm
Speed	510	rev/min
Mep	25,1	bar
Mean piston speed	9,5	m/s
Cylinders	8	in-line
Output/cylinder	720	kW
Power range, mcr	4500	kW
Specific fuel consumption	183-185	$g/kWh$

Table 4: Main parameters of ZA40S engine. (Woodyard 2004)

The inertia of the engine is found from the torsional vibration calculations provided by Geir Dahler in DNV GL, and is added into one single inertia for simulation purposes.

#### 4.2.4 PI-regulator

The control system for the engines is modeled as a PI-regulator as in Equation 10. The regulator has a filter that ensures that the control signal from the regulator never surpass the upper limit set to 1.1, as the engine is not capable of producing more thrust than the maximum. In the simulator the lower limit is set to 0.1, as the assumptions for the simulation is that the engine is always on and Oden moves forward at a constant pace.

#### 4.2.5 Flexible coupling

Flexible couplings are used to transmit torque and rotational velocity from one shaft to another. They can make up for small misalignments, and damp vibrations, shock loads, and noise. In the case of Oden, there is a flexible coupling between the engines and the gearbox. This flexible coupling is a Vulcan Rato 4010 coupling, with unknown characteristics. Therefore, for simulation purposes, the parameters of a similar flexible coupling are used.

The characteristics of the flexible coupling are as follows:

Stiffness	500	$KNm/rad$
Damping	3404	$Nms/rad$

Table 5: Characteristics for the flexible coupling

#### 4.2.6 Gearbox

The gearbox is connected to two engines on one side and to a shaft to the propeller on the other side. Thus, the simplest way of simulating this is using two rotating cogwheels for the engines and one cogwheel for the propeller. This gives three rotating cogwheels in total, which all contributes with rotating inertia. The model also needs a gear ratio, which indicates the size differences in the cogwheels

Inertia cogwheel 1	153.1
Inertia cogwheel 2	153.1
Inertia cogwheel 3	93.3
Gear ratio	$\frac{138.7}{540}$

Table 6: Characteristics for the gearbox onboard Oden (DNV-GL 2020)

#### 4.2.7 Shaft

The shaft from the propel to the gearbox is a steel rod with a length of approximately 15 meters and a diameter of 720 mm. The shaft reaches from spant 18 to 43, spanning over 25 spant.

Assuming a standard spacing of 600 mm between each span, this gives a shaft length of 15 meters.

#### 4.2.8 Propeller

There are two identical propulsion lines installed onboard Oden. The two propellers have a diameter of 4,5 m, 4 blades, and are of the type ducted LIPS CPP, which can be seen in Figure 14. In the simulation model the propel is simulated with a fixed pitch, as the velocity is constant, thus the characteristics for the propeller will be constant.



Figure 14: The propellers installed on Oden. Photo: Garay (2007)

Table 7 gives an overview of the main characteristics for the propeller.

Diameter	4.5	m
Rotational velocity	139	rpm
Rotatin inertia	28369	$kgm^2$

Table 7: Main parameters for the propeller. (Kramers 2016) (DNV-GL 2020)

As it is assumed that the external loads are distributed evenly on both propulsion line, the Equation 7 can be simplified to:

$$\tau_{prop} = 2 \cdot (T_P - D_R) \quad (19)$$

Where  $\tau_{prop}$  is the propulsion of the icebreaker,  $T_P$  is the thrust produced and  $D_R$  is the drag on the rudder.

The drag on the rudder is modelled as a resistance of the propeller given by:

$$T = L_{coeff} |\omega| \omega \quad (20)$$

Where  $T$  is the torque on the propeller,  $\omega$  is the rotational speed of the propeller,  $L_{coeff}$  is a dimensionless coefficient.

### 4.3 Co-simulation model for the system

This section presents the co-simulation model for the propulsion system onboard Oden. The structure presented in Figure 15 shows the structure intended for the co-simulation and software intended for building the simulator in. Each system should have a dynamic connection, having input and output from the other systems. The framework in CSE should be modeled so that each connected simulator can be replaced without modifying the framework for the co-simulation.

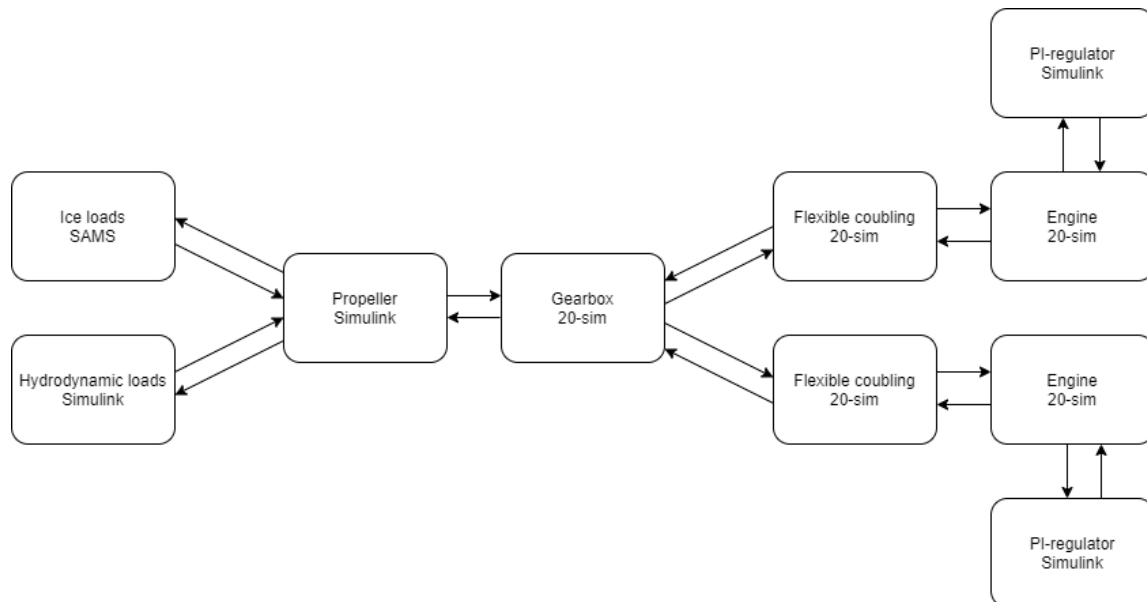


Figure 15: Structure for the co-simulation of one propulsion line with the intended followers and the software used for building the follower.

A dynamic connection to SAMS is not feasible in the current version of SAMS and CSE, SAMS uses TCP/IP connection and CSE uses FMU connection. Connecting Simulink to CSE did also prove to not be feasible. Resulting in that Simulink could not be used as an adapter for the connection between SAMS and CSE. Thus, all simulation models are made using 20-sim, as it was the only software that was compatible. Therefore the ice loads had to come from a pre-run simulation in SAMS. There Oden was moving forward with a constant pace of 3 m/s, hence the assumption of constant velocity for the icebreaker is made for the rest of the simulation. The results from the pre-run simulation in SAMS were processed in MATLAB to make a vector that could be implemented in the 20-sim model **Ice\_loads**. **Ice\_loads** is modeled with an input, which in reality is only a dead end, this to make it compatible with a framework with only dynamic connections and exchangeable for a dynamic ice load simulation model in the future.

The building blocks are created in 20-sim and connected in order to perform a monolithic simulation of one propulsion line. This system can be seen in Figure 16. The model in each block are modeled as described over using either equation models or bond graph.

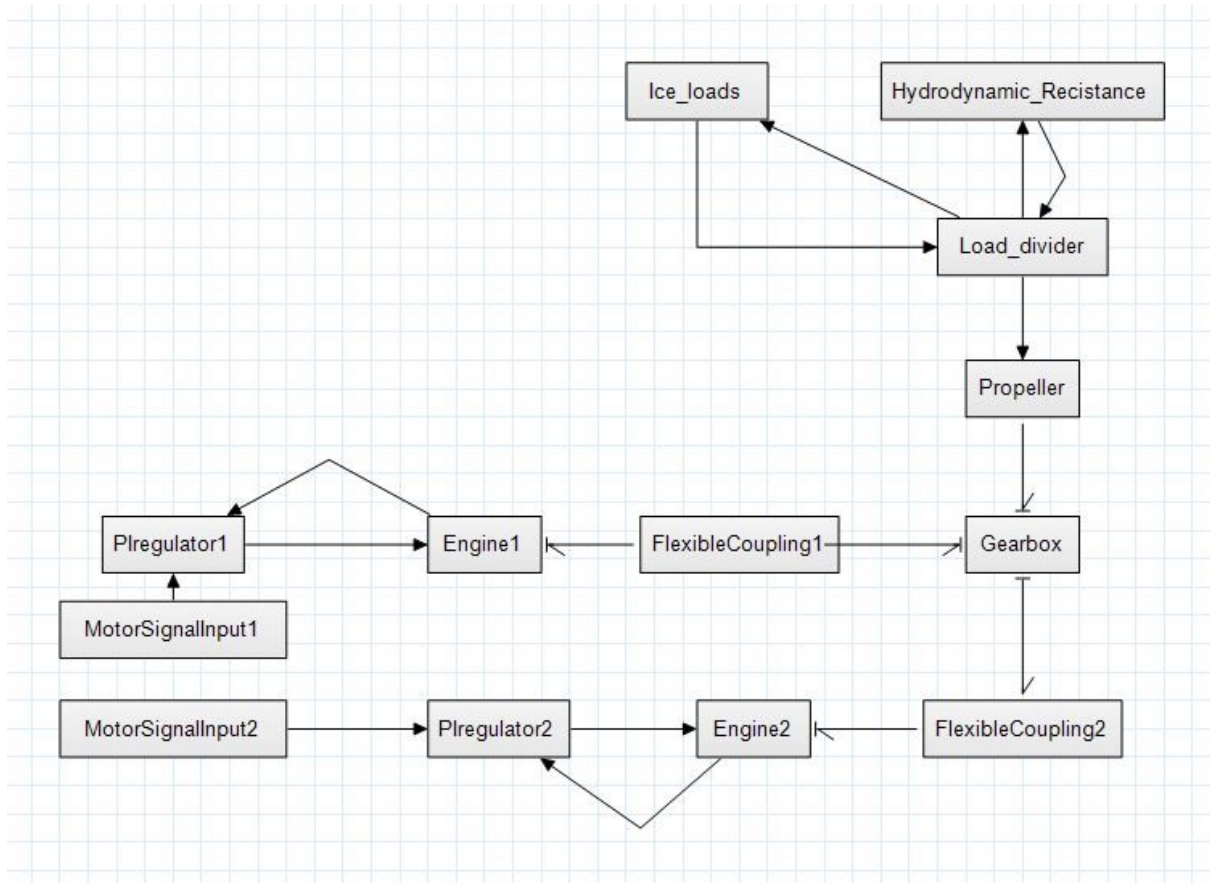


Figure 16: Structure of the monolithic simulation model built in 20-sim

In Figure 16 and Figure 17, arrows are connecting the simulation blocks. The solid arrows indicate one-way signal communication, these are used for input to the propulsion model in terms of ice loads and hydrodynamic resistance on hull, and input and output from the PI-regulators. The half arrows are bond graph arrows, indicating two-way communication in terms of flow and effort. The causal stroke indicates if flow or effort is the input to the block.

In order to simplify the co-simulation model, the structure in Figure 16 is simplified to three blocks that are going to become the followers for the co-simulation model. These three blocks can be seen in Figure 17. Where **Ice\_Loads** and **Hull\_Resistance** are the same as before, and the rest of the simulation model is in **Propulsion\_System**. **Propulsion\_System** contains only one propulsion line, therefore there block **Load\_divider** is included.



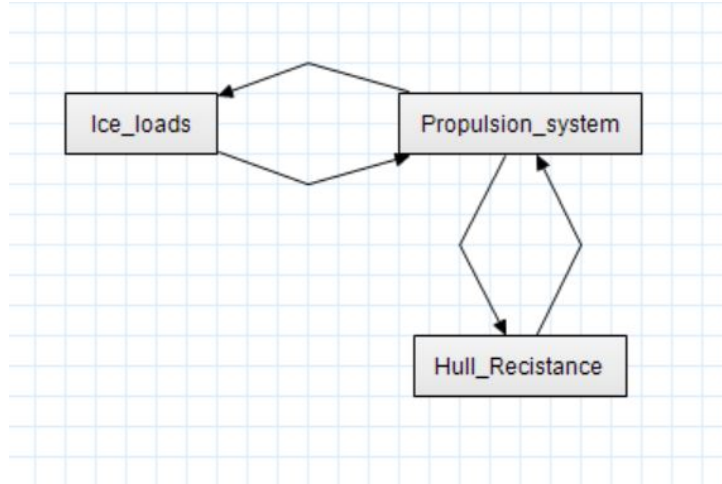


Figure 17: Simplified structure for the co-simulation

The three followers in Figure 17 is exported using the FMI 2.0 Co-simulation Export. They are reassembled to a co-simulation in the files *SystemStructure.ssd* and *OspSystemStructure.xml*, they can be seen in Appendix 2 and Appendix 3 respectively. These files contain a list of where to find the FMUs, communication frequency, and how each signal is connected with each other.

<b>Start</b>	<b>Out port</b>	<b>End</b>	<b>In port</b>
Ice_Loads	output	Propulsion_system	IceLoadIn
Hull_Resistance	output	Propulsion_system	HullResistance
Propulsion_system	velocityIce	Ice_loads	input
Propulsion_system	velocityHull	Hull_recistance	velocity

Table 8: Co-simulation followers and contact points

## 5 Results

### 5.1 Input to co-simulation model

The input to the co-simulation is the results from a pre-run simulation of Oden moving at a constant pace of 3 m/s through an ice field in SAMS. These results was used to calculate the forces from breaking the ice, the ice rubble and the total ice load, this was done in MATLAB using equation Equation 17 and Equation 18. A complete graph showing the icebreaking force, ice rubble forces and total ice load on the hull can be seen in Appendix 1.

The following three figures shows the ice loads experienced by Oden in the period of time that the co-simulation run in CSE. Figure 18 shows the load occurring as Oden breaks the ice in front of her.

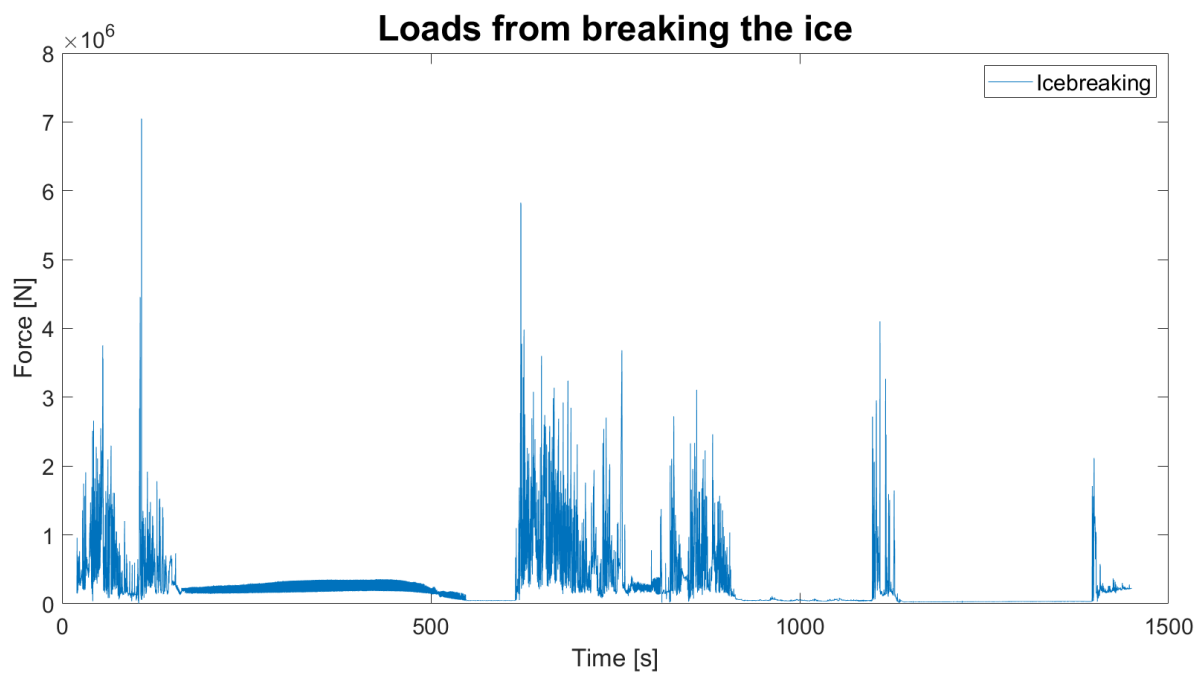


Figure 18: Forces from breaking the ice

Figure 19 show the loads occurring on the hull of Oden as the broken pieces of ice slides along the hull of Oden.

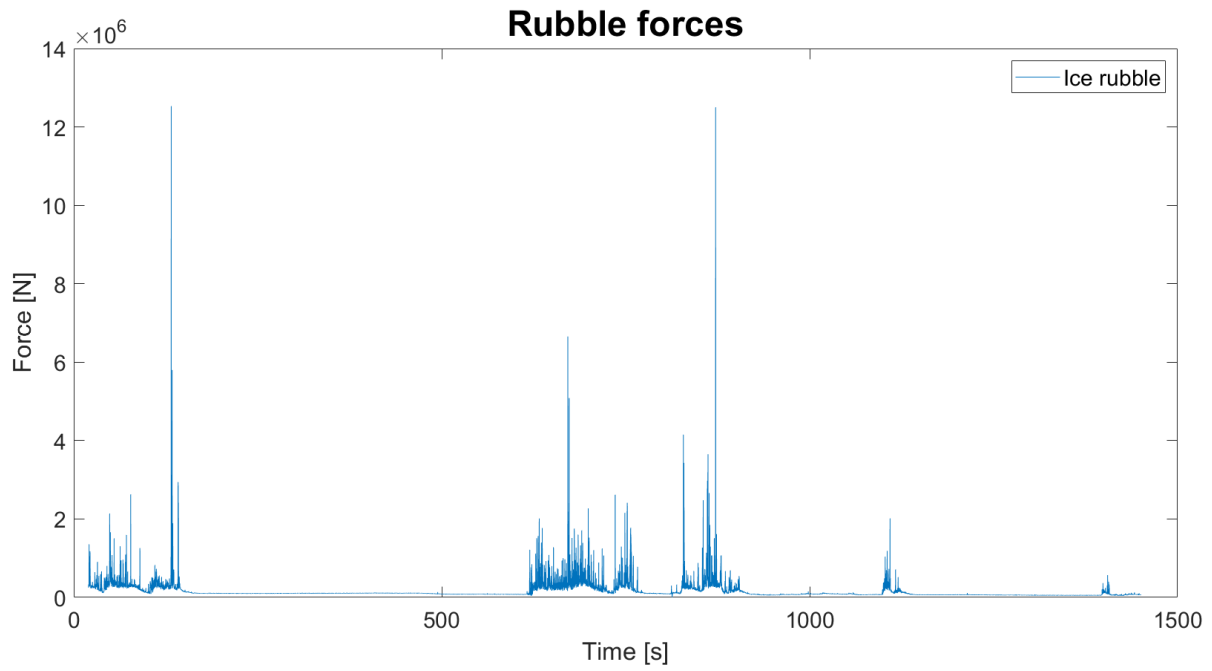


Figure 19: Forces from ice rubble sliding and slamming along the hull of Oden

These forces are added together to give the total load from the ice hull interactions, which are shown in Figure 20. This load is the input to the co-simulation and can be found in the folder **Ice\_Loads**.

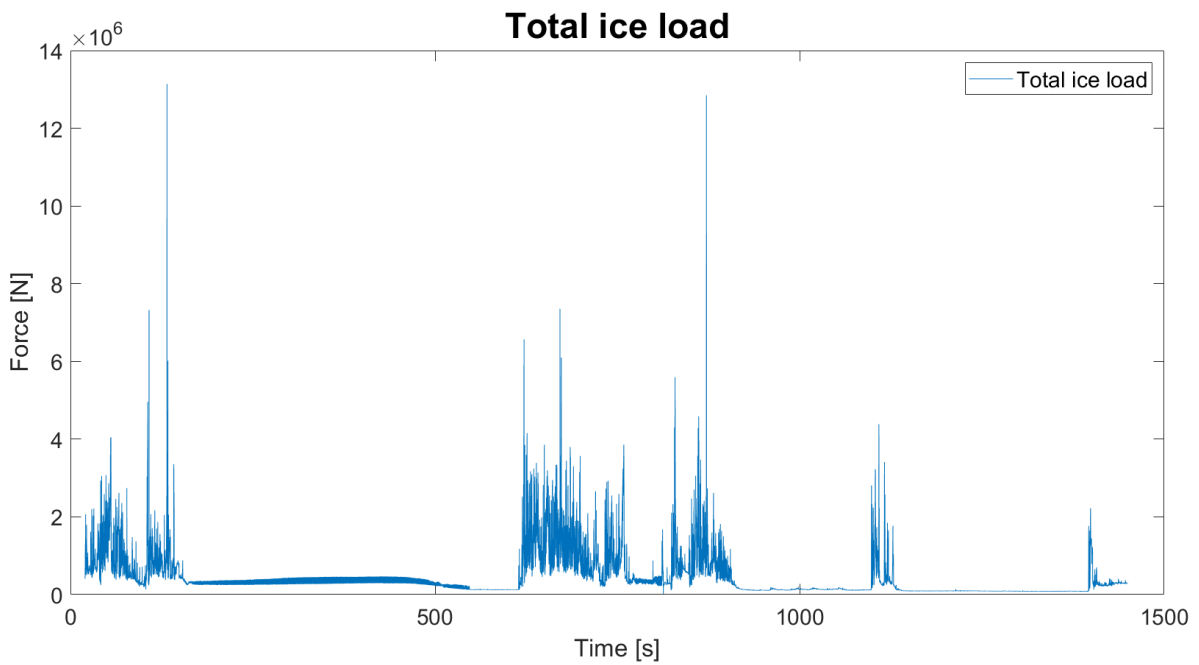


Figure 20: Total ice load on the hull, which consist of the icebreaking and rubble force combined.

## 5.2 Results from the co-simulation

Figure 21 shows the input data to the propeller. Where the spiking orange line is the sampled ice load, from Figure 20. Hull resistance is shown as a constant blue line, which is calculated in the simulation model **Hull\_Resistance**. The hydrodynamic resistance on the hull is calculated based on the velocity as shown in Equation 16. As the velocity of Oden is constant, thus the hydrodynamic resistance on the hull becomes constant.

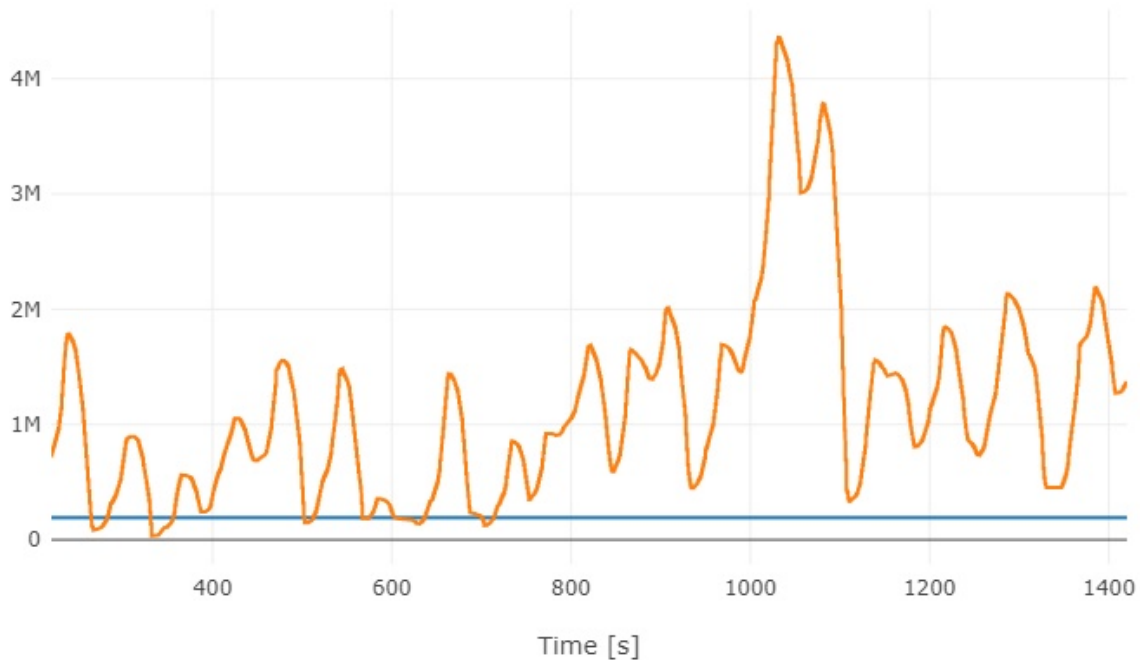


Figure 21: Load input to the propulsion system. The blue line is hydrodynamic resistance on the hull and the orange is the total ice load on the hull. Unit on y-axis is Newton

The blue line on Figure 22 shows the total external force on Oden, it is the sum of the hydrodynamic resistance and ice loads on the hull, which are shown on Figure 21. The orange line on Figure 22 shows the external load on one propeller. It is assumed that the load from the external forces acting on the hull is distributed equally on both propulsion lines, thus the external forces are divided in half and only one propulsion line is simulated.

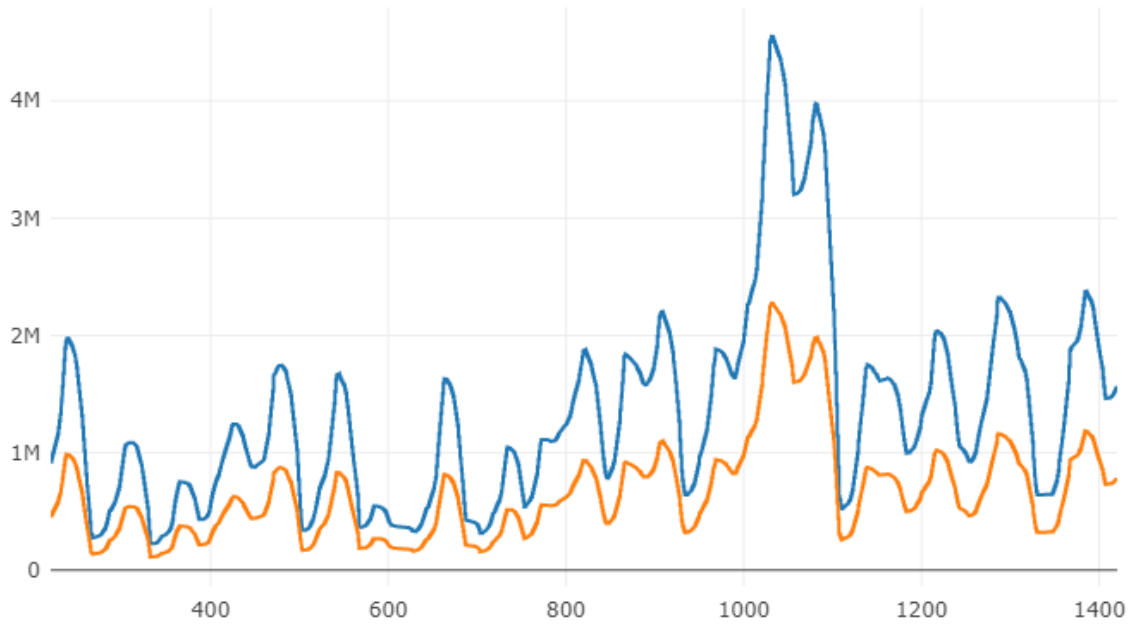


Figure 22: Total load on hull versus load on one propeller. The blue line the total external force and the orange is the load on one propeller. Unit for the x-axis is [s] and y-axis is [N]

Figure 23 shows the external load on one propeller, it is the same plot as in Figure 22, but with adjusted axis to show more details.

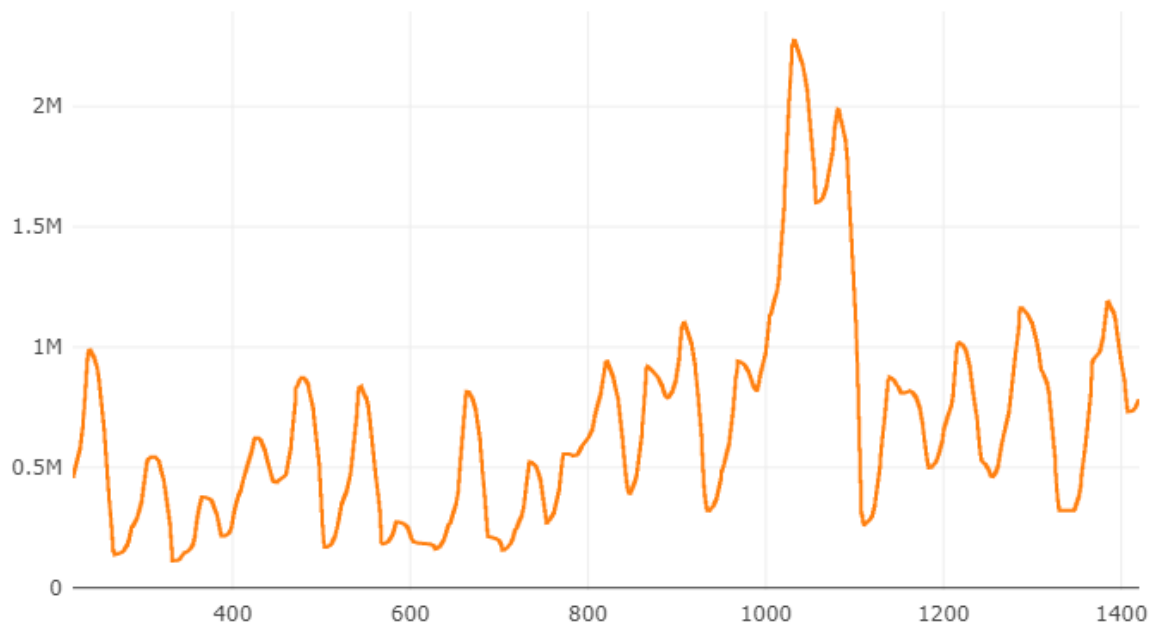


Figure 23: External load on one propeller. Unit for the x-axis is [s] and y-axis is [N].

Figure 24 shows the rotational velocity of the propel.

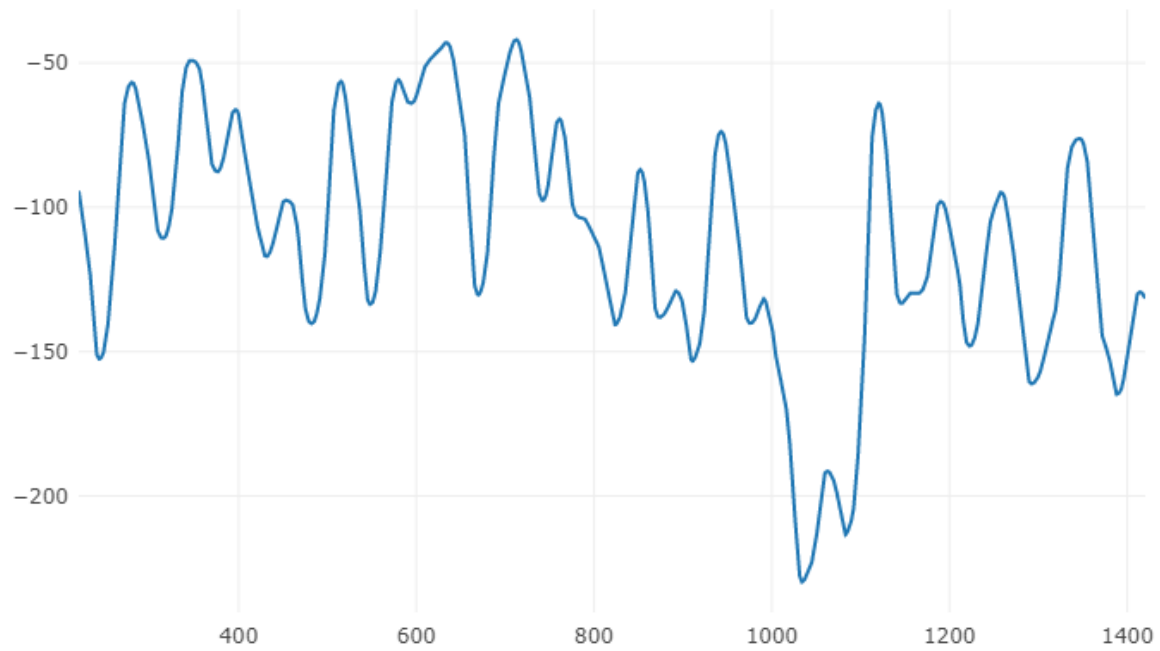


Figure 24: Rotational velocity on the propeller. Unit for the x-axis is [s] and y-axis is [rad/s].

Figure 25 shows the resistance on the propel. The plot of the angular velocity and propeller resistance has a similar shape as the resistance is calculated from the angular velocity as shown in Equation 20.

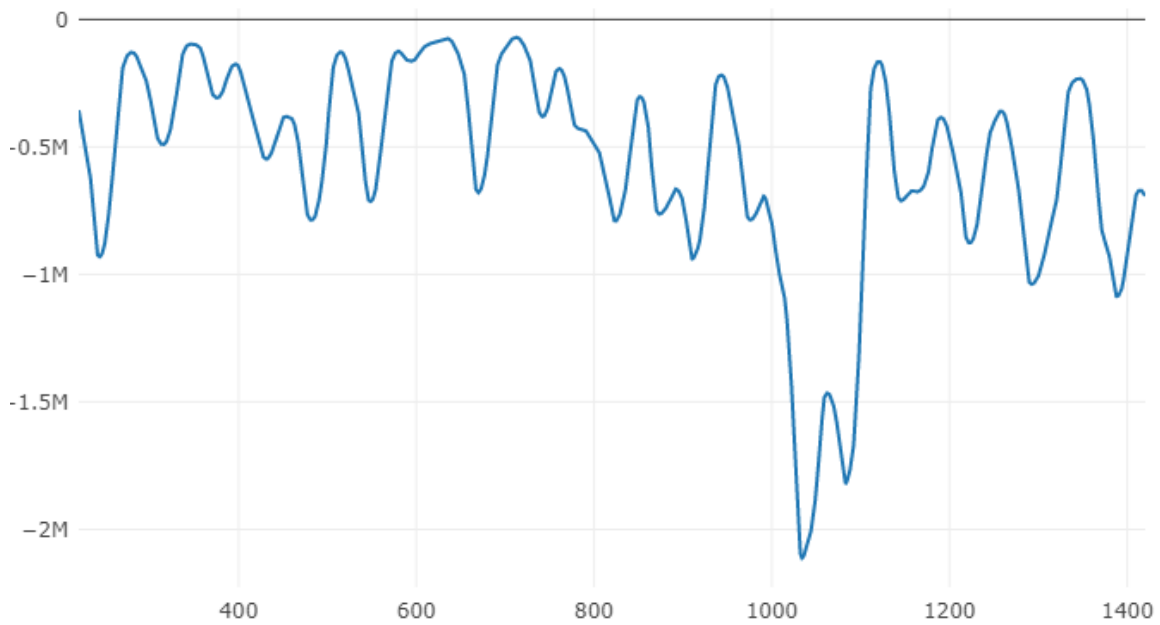


Figure 25: Resistance on propeller. Unit for the x-axis is [s] and y-axis is [Nm].

Figure 26 show the angular velocity in and out of the gearbox. The orange line shows the angular velocity between the gearbox and flexible couplings on both connections as the angular velocity is identical on both. The green line is the rotational velocity on the outlet of the gearbox towards the propeller.

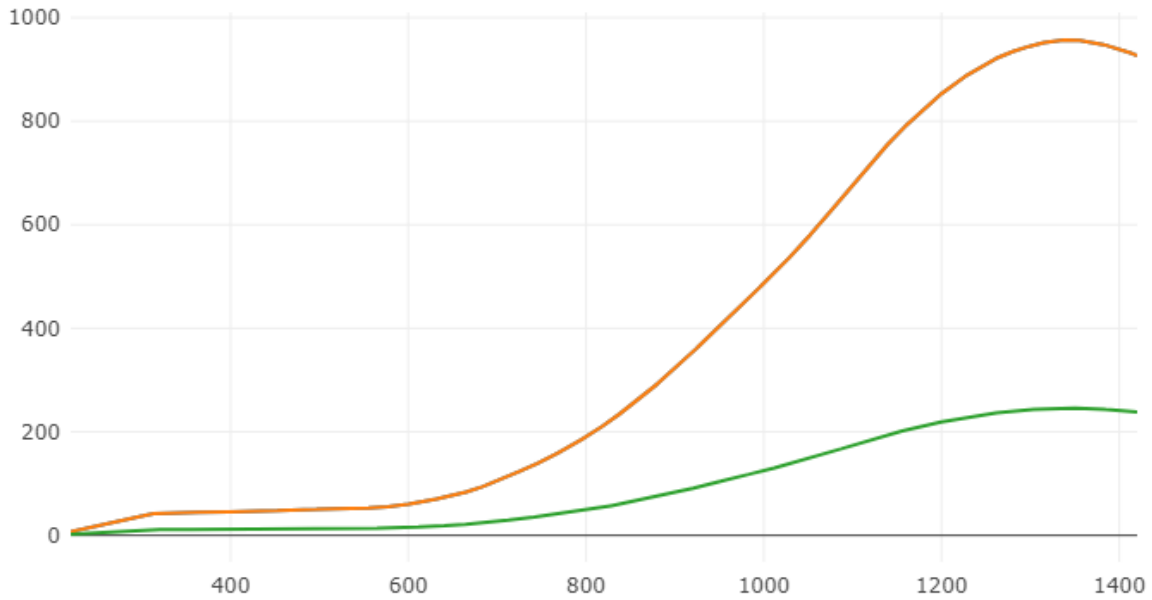


Figure 26: Angular velocity over the gearbox. Unit for the x-axis is [s] and y-axis is [rad/s].

Figure 27 shows the torque on the gearbox. The blue line is the torque where the gearbox is connected to flexible coupling 1 and the green line is the torque where the gearbox is connected to flexible coupling 2. The orange line shows the torque from the gearbox towards the propeller, this is small relative to the other side of the gearbox, and can be seen in more detail on Figure 28.

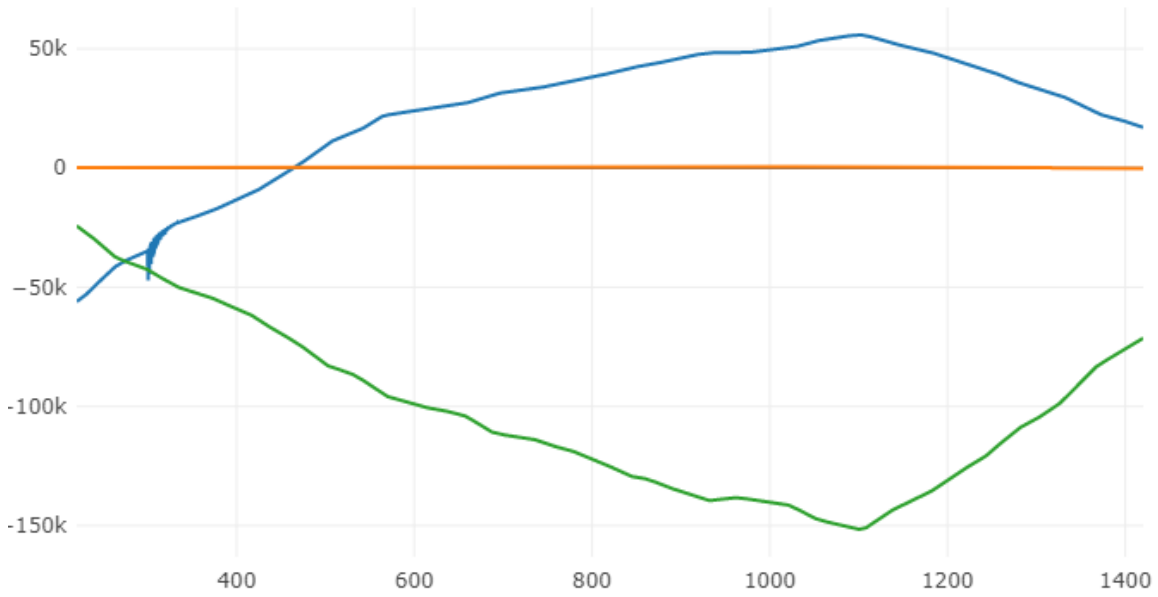


Figure 27: Gearbox torque. Blue and green goes towards the engine 1 and engine 2, and orange towards propel. Unit for the x-axis is [s] and y-axis is [Nm].

The orange line on Figure 28 shows the torque out of the gearbox on the propeller side, which is the same as on Figure 27 but with different scaling on the axis.

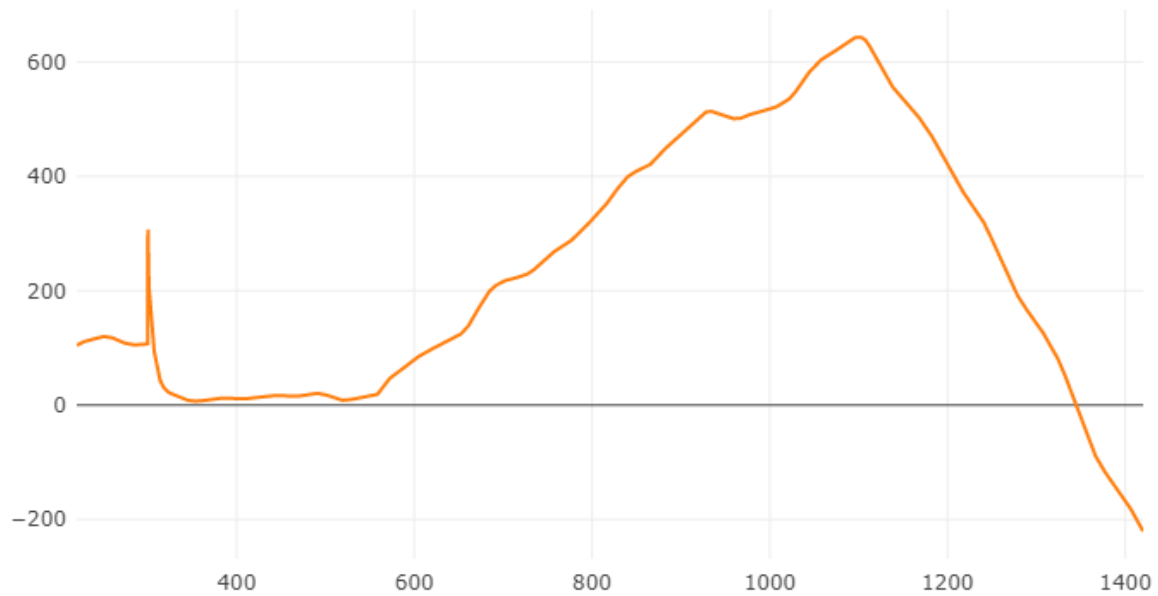


Figure 28: Gearbox torque, only the outlet towards propeller. The orange line is the same as on Figure 27, but with different scale on the axis. Unit for the x-axis is [s] and y-axis is [Nm].

Figure 29 shows the torque delivered by the engines. The blue line shows that engine 1 is



delivering torque at a constant level. While engine 2, the orange line, starts with delivering torque which gradually goes toward minimum torque deliverable.

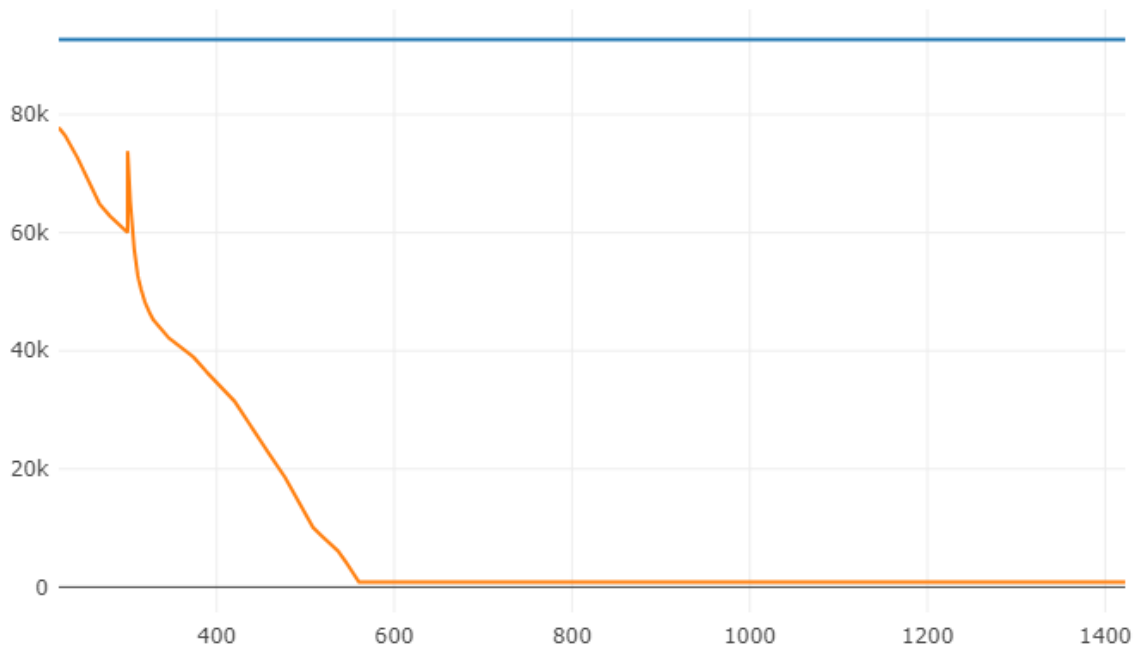


Figure 29: Torque delivered by the engines. Blue line is engine 1 and orange line is engine 2. Unit for the x-axis is [s] and y-axis is [Nm].

Figure 30 show the rotational velocity of the engine, which is the input to the engine controller. The green line is the rotational velocity on engine 1 and the red line is on engine 2.

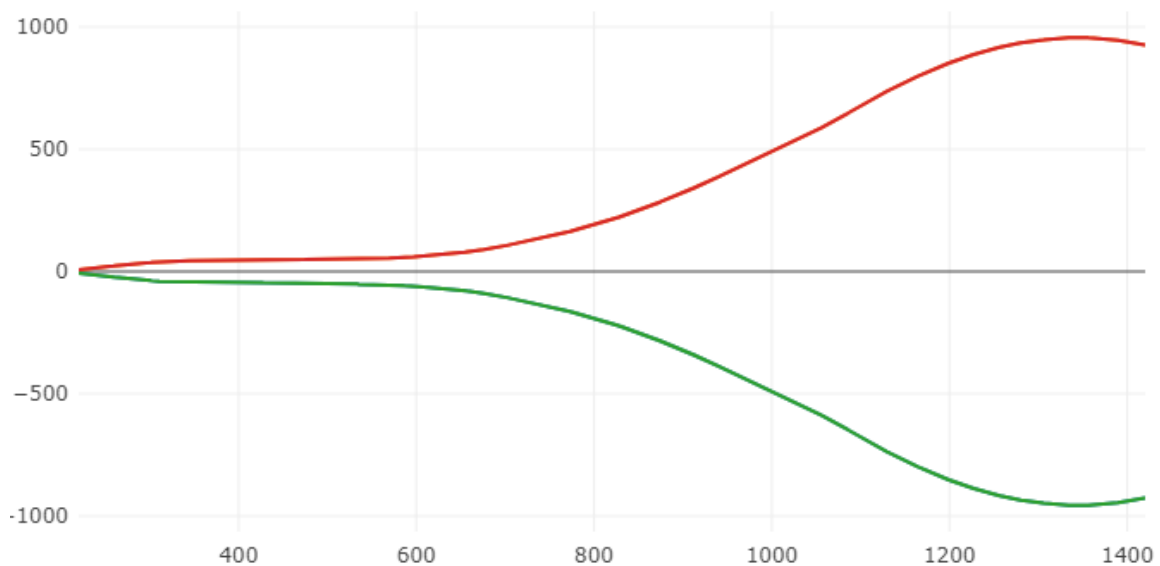


Figure 30: Plot shows the rotational velocity of the engines. The green line is engine 1 and the red is engine 2. Unit for the x-axis is [s] and y-axis is [rad/s].

Figure 31 shows how far each engine is from the control set point, shown as an orange line. The green line is the error for engine 1 and the red line is the error for engine 2. The set point for the rotational velocity on the engines are 8,5 rad/s.

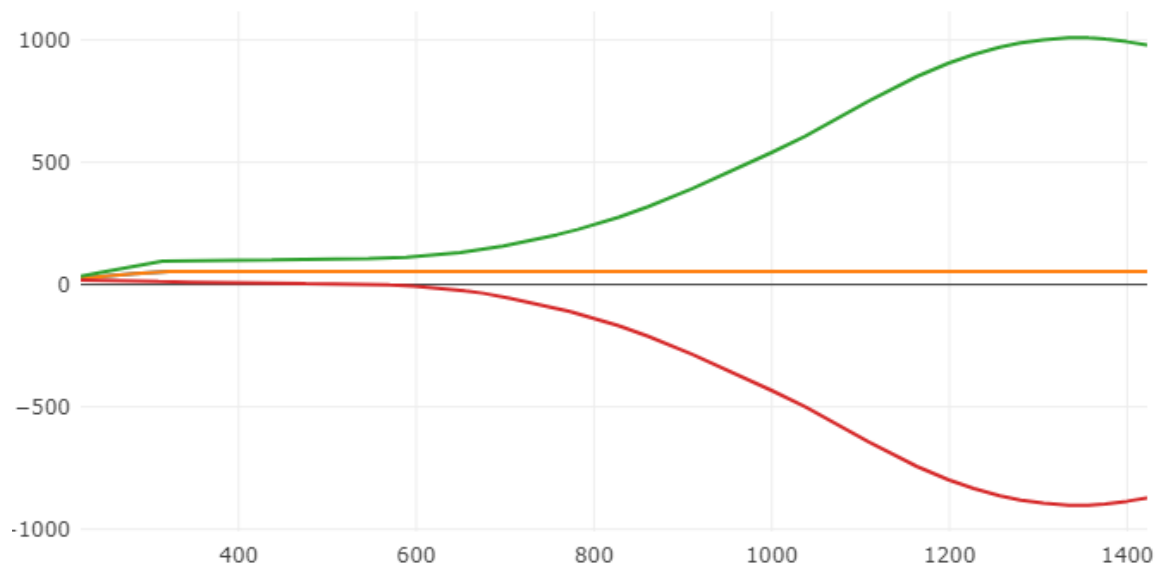


Figure 31: The difference between the rotational velocity of the engines and the controller set point. The orange line is the controller set point for the engines. The green line is the error for engine 1 and the red line is engine 2. Unit for the x-axis is [s] and y-axis is [rad/s].

Control signal from the PI-regulators, can be seen on Figure 32. The red line is the control signal from PI-regulator 1 and blue line is PI-regulator 2.

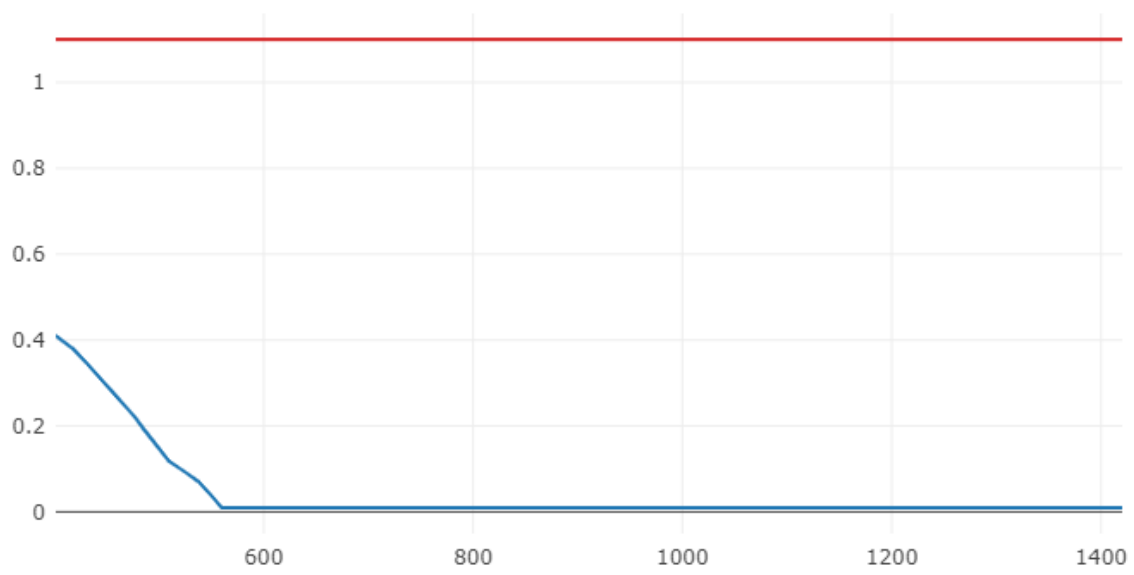


Figure 32: Control signal from the PI-regulators. The red is PI-regulator 1 and blue is PI-regulator 2. Unit for the x-axis is [s] and y-axis is [-].

## 6 Discussion

### 6.1 Choice of DSB

The initial part of the process of developing the digital twin for the propulsion system onboard Oden was investigating the current state of the art and familiarize with the prototype software CSE. CSE is developed as an open-source collaborative co-simulation software. CSE is a tool for collaboration between different companies, thus the software has to be created in such a manner that it protects collaborators code, business secrets in the simulation models, etc. To achieve that CSE has to be tested by independent parties to determine if it works is convenient, and fulfills the needs of the user.

Simulink can function as the co-simulation software and is a tool that works with SAMS as it has the TCP/IP connection and FMU functionality. This makes Simulink versatile and a good fit to solve the model fitted.

Thus, the questions remain; why not switch the co-simulation environment from CSE to Simulink? The reason for sticking with CSE although Simulink would be capable of simulating more of the issues related to how the ice impacts the systems onboard the icebreaker, is that CSE is free, open-source, and made for collaboration. Even though Simulink is capable, has its downsides in terms of availability, to do the co-simulation using FMUs in Simulink, an available license for the host and an available license for each connected FMU is necessary. This can be very expensive compared to the free open-source alternative CSE.

### 6.2 Simulation tools for the followers

As CSE is a tool for developing only the framework for the co-simulation, the connected simulation models need to be developed using other simulation tools. Thus, the initial phase of the project was to find suitable software for the connected simulation model, meaning software for followers. The requirement was that the simulation software could be exported as an FMU and implemented in the CSE framework. FMI-standard.org has a list of software which can export simulation models as FMUs which are suitable for co-simulation (FMI-standard.org 2020). Out of these software, 20-sim and MATLAB Simulink are available to NTNU students and suitable for building simulation models. There also exist several extensions for Simulink to make FMU export possible.

Both 20-sim and Simulink were tested for use in CSE. From these tests, it was found that 20-sim is suitable for building simulation models, capable of exporting FMUs and that those FMUS could be implemented in CSE. Simulink was also found suitable for building simulation models and FMU export for co-simulation purposes, but only if the co-simulation environment is Simulink and not CSE. As CSE is the chosen co-simulation environment in this thesis, Simulink is therefore unfit for building the connected simulation models in this case.

At the time which it was discovered that Simulink could not be used on the current state of the development of CSE, the idea of using Simulink as an adapter for SAMS was rejected. It was known that simulation models in SAMS can not be exported as FMUs, but uses TCP/IP

to connect to other software. The idea was to connect SAMS to Simulink and then export the connection as an FMU to insert in the CSE framework. This was never tested, as Simulink was excluded as a possibility. To connect SAMS to CSE in the future, either SAMS must be updated with an FMU export for the co-simulation feature or the connection fault between CSE and Simulink should be fixed, for Simulink to work as an adapter.

### **6.3 Monolithic or distributed simulation model**

As all followers were built in the same simulation software, thus the simulation model can be simulated as one complete model there. Additionally, the compatibility and dynamics between each follower can be checked before exporting the model. Although a complete analysis of the system is possible in 20-sim and the extra steps of exporting the model and then reassemble it using CSE does not give any additional knowledge about the system analyzed, this process seems unnecessary. Particularly because some data can be lost in this process, and it is more time consuming than just sticking with one software. However, this thesis intends to test the capabilities of, and exploring the co-simulation software CSE and also begin the process of building a digital twin of Oden which can be further developed and extended later on. This purpose is the reason for the extra effort put into the model, as it will give knowledge about the possibility that exists in the co-simulation and there is a system ready for expansion and inclusion of other simulation models.

### **6.4 Inclusion of the ice loads**

The initial idea of having a dynamic connection between SAMS and the remaining simulation model by using Simulink as an adapter did not work as intended, as CSE and Simulink did not cooperate. Thus the ice loads had to be implemented in the simulation model in another way. This was done by using the pre-recorded data set from SAMS, with Oden moving at a constant pace of 3 m/s. It was tested to implement the ice loads in 20-sim by reading from a txt-file, which worked, but not when exporting the model as an FMU. The third solution for implementing the ice loads in the model was to make a vector of the ice loads in 20-sim, which worked also when exporting the model.

The vector of ice loads contains the total ice load calculated from the ice rubble forces and the icebreaking forces in x- and y-direction as described in subsection 4.2.2. This gives the total load the vessel has to overcome in order to move forward in the ice field. The total data set has more than 900,000 recordings, which is too big to implement in 20-sim. From testing different lengths for the time series included as a vector, it was found that 70000 data points is within the limit and 72000 exceeds the limit, it was not tested any further to find the exact limit as 70000 data point gives a more than sufficient length of the time series.

A pre-recorded ice load where the vessel moves at a constant pace, gives an unrealistic picture of the ice loads. This is due to that the impact from the ice on the hull depends on the velocity of the vessel and the rapid changes in the load magnitude influences the velocity of the vessel. In reality, there would be a dynamic relationship between the velocity of the vessel and the ice load on the hull, this is not reflected in this simulation model. The chosen velocity for the simulation

is the maximum speed that Oden is capable of keeping in level ice of thickness 1.9 m.

## 6.5 Consequence of assumption

The list below gives a summary of the assumptions made for the simulation model.

1. The icebreaker propagates through the ice at a constant speed of 3 m/s. This indicates that the acceleration must be zero.
2. The external forces that act on the vessel are limited to hydrodynamic forces and ice loads, all other external forces are neglected.
3. Only one propulsion line is simulated as the loads are evenly distributed on both lines, thus the external loads are divided in half.
4. The engines are always on and are always producing positive torque.

As there was found several limitations in how the co-simulation could be conducted, and the assumption regarding constant velocity is a consequence of that. This is a unrealistic assumption, as Oden would not be capable of keeping maximum speed with no fluctuation in the velocity in an ice field with irregular ice loads.

In the co-simulation the environmental loads are limited to the resistance on the hull of icebreaker and ice loads, which are the major loads. But a more accurate representation of the environmental loads would included wind, current and wave loads.

In the co-simulation model only one simulation line is included as it is assumed that the external loads distributes equally on both propulsion lines. This is a simplification, which is not necessary correct as ice can get stuck in one propeller and cause additional loads.

## 6.6 Uncertainties regarding the parameters used in the simulation

The accuracy of the simulation model depends on the provided input data to the model, as for the simulation model of the propulsion system onboard Oden the internal inertia is based on the TVC. This calculation was performed in 1987, and lacks some information, e.g. the flexible coupling between engine and gearbox is a Vulcan Rato 4010 coupling, but the stiffness and damping of the given coupling are not known. Therefore one has to make an assumption regarding the characteristics of this coupling. The gearbox is produced by a company named RENK-AG, but the specific model of the gearbox is unknown. For the gearbox not all inertias are known, neither is the exact gear ratio. The gear ratio used for the simulation is calculated from the maximum input RPM and the resulting RPM on the propeller. There will be some minor errors, which are that there will be losses throughout the engine, shaft, and coupling before the gearbox and after the gearbox until the propeller. To find the exact gear ratio one can do measurements over the gearbox.

## 6.7 The results from the co-simulation

The co-simulation model was successfully implemented in CSE with the structure shown in Figure 17. The followers were tested for interchangeability, and that did work according to plan.

The input to the propulsion system can be seen in Figure 21. Comparing this orange line for ice loads with the ice load seen in Figure 20, they are visibly different. This is a consequence of the vector implementation of the ice loads. Even though this ice load varies slower than in the input file, it still has the characteristic spiking load that is typical for ice-hull interactions when an icebreaker propagates in ice. Thus the resulting ice load can still be used to test the co-simulation and give an impression on how fluctuating ice loads impacts the propulsion system.

Given the input to the co-simulation, the total load on the hull was calculated, and divided in half, this can be seen in Figure 22 and in more detail on Figure 23. This was done as it was assumed that the global load acting on the icebreaker would have an equal impact on both propulsion lines at all times. This is simplification done in the simulation, as it is not known how the external loads are distributed between the propulsion lines. On the physical icebreaker, the loading on each propulsion line will be uneven as the ice floes will hit the icebreaker on random places and with random load magnitudes. The load distribution between the propulsion line will change with the heading and velocity of the icebreaker. The consequence of doing this is that there might be higher loads that are not encountered for.

Figure 24 shows the rotational velocity of the propeller, which is of a larger magnitude than expected and goes in a negative direction. The setpoint for the engine is set such that the output on the propeller would ideally become 139 rpm, which equals 14.5 rad/s. The magnitude and direction of the rotational velocity are a consequence of is going on with the engines, but as the engines do not behave as intended the fault there propagates to the propeller.

The resistance on the propeller corresponds with the rotational velocity of the propeller. The plot in Figure 25 shows the resistance on the propeller.

The angular velocity over the gearbox shown in Figure 26, shows the expected rotational direction, but the magnitude is larger than expected.

Figure 27 and Figure 28 show the torque over the gearbox, here the green and blue line shows the torque where the engines are connected with flexible couplings to the gearbox. The orange line shows the torque towards the propeller. The behavior in the gearbox is strange, it was not expected to have negative torque in the simulation model. In the 20-sim model for the gearbox, there is an algebraic loop that may cause faults in the simulation model. It would be recommended to investigate the gearbox to see if the bond graph could be modeled differently, or if a different approach for modeling the gearbox would fix the co-simulation model.

Figure 29 shows that engine 1, the blue line, deliver maximum torque. This is an expected reaction as the PI-regulator register the negative angular velocity and compare it with the set point, which gives a huge error, as seen in Figure 31. Thus the PI-regulator forwards the signal about maximum torque to the engine 1.

On the other side the PI-regulator for engine 2 registers a rotational velocity higher than the set point. Thus the PI-regulator forwards a control signal in order for engine 2 to reduce the velocity, which also happens. The control signal from the PI-regulator to engine 2 can be seen as an orange line on Figure 32.

## 6.8 The new pre-release of the co-simulation software

On the 8 June 2020 the first public release was launched (Open Simulation Platform 2020b). The updated version is named *Open Simulation Platform interface specification*, OSP-IS.

This version is not tested, but by looking through the list of Maritime Reference Models and the tools for them, it seems like the new version works with Simulink FMUs (Open Simulation Platform 2020a). From this list one can see that the new version is tested with FMUs from several software, such as 20-sim, SimulationX, Simulink, Vesim, Open Modelica, C++, Java.

This version was released so far into the process of working on this thesis that building a new simulation model could not be prioritized, therefore the new software is not tested. However, given more time, the current model should be tested in the newest version of the software and the idea of using Simulink as an adapter should also be tested.



## 7 Conclusion

The co-simulation provides opportunities to connect different simulation models to gain new insights through simulations. As an example, one can connect simulation models of ice-hull interactions in SAMS with simulation models of other parts of an icebreaker, thus gaining more knowledge about how ice loads impact systems, like the propulsion line, within the icebreaker.

This thesis aims to establish a co-simulation model of the icebreaker Oden. The goal was to connect the existing ice-hull interaction SAMS model of Oden with a simulation model of the propulsion system. This connection was made with the co-simulation software CSE. The followers for the propulsion line and hull were to be developed in 20-sim, and Simulink. It was found that only 20-sim were compatible with CSE. Thus, all followers had to be developed using 20-sim. Since SAMS could not be connected with CSE, ice loads from a pre-run simulation in SAMS of the icebreaker Oden moving at a constant pace of 3 m/s were used instead.

The co-simulation model is not in a sufficient condition that it could be used for analyzing the impact ice loads have on the propulsion line. The resulting plots from the co-simulation shows that the simulation model of the engines and the gearbox does not behave as expected, as there are negative torque and rotational velocities. This is most likely due to an algebraic loop in the gearbox model which causes faulty behavior in the co-simulation model. There was not adequate time to solve this satisfactorily.

Throughout the work on this thesis, it became evident that this project began too early in the development of the co-simulation tool provided by the Open Simulation Platform. Most of the time-consuming trial and error could be avoided, if the project was initiated when the finished version of the co-simulation software was released. Many of the obstacles that were faced during the development of the co-simulation model would not be present in the new version of CSE, based on the information about the software and the Maritime Reference Models.

## 7.1 Recommendation for further work

Working on this thesis, there has been a lot of trial and error, following is a list of recommendations for future improvements of the co-simulation model.

- Firstly, the simulation model of the gearbox should be fixed, this contains an algebraic loop that should be removed.
- The co-simulation should be done in the updated version of CSE, OSP-IS. In this version the co-simulation should be tested using followers from Simulink.
- In order for SAMS to be part of a co-simulation it needs a working connection to the co-simulation software. As SAMS and CSE have different types of connections, TCP/IP and FMU respectively, one of them need a new type of export feature. This is also true for the upgraded version of CSE, OSP-IP.

## Bibliography

- Blochwitz, T., Otter, M., Arnold, M., Bausch, C., Elmqvist, H., Junghanns, A., Mauß, J., Monteiro, M., Neidhold, T., Neumerkel, D. et al. (2011), The functional mockup interface for tool independent exchange of simulation models, *in* ‘Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany’, number 063, Linköping University Electronic Press, pp. 105–114.
- Controllab (2020), ‘What is 20-sim?’. Accessed July 27, 2020.  
**URL:** <https://www.20sim.com/>
- DNV-GL (2020), ‘Sulzer - Torsional vibration of the shafting’. Contact DNV-GL for details.
- FMI-standard.org (2020), ‘Tools’, <https://fmi-standard.org/tools/>. Accessed July 28, 2020.
- Fossen, T. I. (2011), *Handbook of marine craft hydrodynamics and motion control*, John Wiley & Sons.
- Garay, L. (2007), ‘8 October 2007 Oden Southern Ocean 2007 Workshop’. Accessed July 28, 2020.  
**URL:** <https://www.polartrec.com/expeditions/oden-antarctic-expedition-07/journals/2007-10-08>
- Gartner, Inc. (2018), ‘Gartner Identifies Five Emerging Technology Trends That Will Blur the Lines Between Human and Machine’. Accessed February 5, 2020.  
**URL:** <https://www.gartner.com/en/newsroom/press-releases/2018-08-20-gartner-identifies-five-emerging-technology-trends-that-will-blur-the-lines-between-human-and-machine>
- Heyn, H.-M. (2019), ‘Motion sensing on vessels operating in sea ice: A local monitoring system for transit and stationkeeping operations under the influence of sea ice’.
- Intelligent Systems Lab @ NTNU in Ålesund (MechatronicsLab) (2020), ‘MAROFF KPN: Digital Twins for Vessel Life Cycle Service (TwinShip) (2018-2021) NEW FUNDED!’, <http://mechatronics.hials.org/project/project-kpn/>. Accessed July 24, 2020.
- Kramers, J. (2016), ‘Global ice ridge ramming loads based on full scale data and specific energy approach’, p. 244.
- Kulyakhtin, A. (2018), Example of icing, conditions for marine spray icing. Lecture Notes from AT-327.
- Kyllingstad, L. T. (2018), ‘Virtual Prototyping of Maritime Systems and Operations: the project’.  
**URL:** <https://viproma.no/doku.php?id=viproma:about>
- Kyllingstad, L. T. (2019), ‘Open Simulation Platform, Co-simulation, standards and software’.
- Larsson, L. & United States Navy (2008), ‘Swedish icebreaker oden’. Accessed July 24, 2020.  
**URL:** <https://www.navy.mil/management/photodb/photos/080407-O-xxxxX-004.jpg>

- Linder, C. (2008), ‘Aboard the Swedish Icebreaker Oden’, <https://www.wired.com/2008/01/ff-arctic-ss/>. Accessed July 27, 2020.
- Løset, S., Shkhinek, K. N., Gudmestad, O. T. & Høyland, K. V. (2006), *Actions from ice on arctic offshore and coastal structures*, LAN St. Petersburg, Russia.
- Lubbad, R., Løset, S., Lu, W., Tsarau, A. & van den Berg, M. (2018), ‘An overview of the Oden Arctic Technology Research Cruise 2015 (OATRC2015) and numerical simulations performed with SAMS driven by data collected during the cruise’, *Cold Regions Science and Technology*.
- Ludvigsen, K. B. (2019), ‘Open Simulation Platform, Building digital twin systems and ships - together’.
- MathWorks (2020), ‘What is a digital twin?’. Accessed July 24, 2020.  
**URL:** <https://se.mathworks.com/discovery/digital-twin.html>
- NASA’s Scientific Visualization Studio; AMSR2 data courtesy of the Japan Aerospace Exploration Agency (JAXA); The Blue Marble data is courtesy of Reto Stockli (NASA/GSFC) (2019), ‘Arctic sea ice minimum 2019’, <https://svs.gsfc.nasa.gov/4757>. Accessed July 24, 2020.
- Open Simulation Platform (2019), ‘Osp core simulation environment released for testing’.  
**URL:** <https://opensimulationplatform.com/2019/05/06/osp-core-simulation-environment-released-for-testing/>
- Open Simulation Platform (2020a), ‘Maritime Reference Models’, <https://open-simulation-platform.github.io/demo-cases>. Accessed July 28, 2020.
- Open Simulation Platform (2020b), ‘open-simulation-platform/cosim-demo-app/releases’, <https://github.com/open-simulation-platform/cosim-demo-app/releases>. Accessed July 28, 2020.
- Pedersen, E. & Engja, H. (2014), *Mathematical Modelling and Simulation of Physical Systems Lecture Notes in Course TMR4275 Modelling, Simulation and Analysis of Dynamic Systems*, Technical report, Department of Marine Technology Norwegian University of Science and Technology.
- Rindarøy, M., Nielsen, J. B., Hassani, V., Fathi, D., Skjong, S., Sadjina, S., Pedersen, E. & Kyllingstad, L. T. (2015), *Virtual Prototyping Framework Specification*, MARINTEK.
- Riska, K. (2011), ‘Design of ice breaking ships’, *Course material NTNU 114*.
- Sanderson, T. J. O. (1988), *Ice mechanics : risks to offshore structures*, Graham Trotman, London.
- Shankland, S. (2020), ‘Twitter engineers replacing racially loaded tech terms like ’master,’ ’slave’’.  
**URL:** <https://www.cnet.com/news/twitter-engineers-replace-racially-loaded-tech-terms-like-master-slave/>

- Skjong, S. (2017), 'Modeling and Simulation of Maritime Systems and Operations for Virtual Prototyping using Co-Simulations', **2017**(December), 301.
- Smithson, P., Addison, K. & Atkinson, K. (2002), *Fundamentals of the physical environment*, 3 edn, Routledge, chapter 24 Polar environments, pp. 481–489.
- Stensvold, T. (2016), 'Trendene på verdens største skipsmesse: – «Digital tvilling» av skip og nye drivstoffløsninger'.
- URL:** <https://www.tu.no/artikler/dnv-sjef-om-trendene-pa-verdens-storste-skipsmesse-digital-tvilling-av-skip-og-nye-drivstofflosninger/351209>
- Swedish Maritime Administration (2019), ICEBREAKER ODEN, Technical report. Accessed October 20, 2019.
- URL:** <http://www.sjofartsverket.se/pages/41381/Oden%20f%C3%B6r%20webben.pdf>
- Tsarau, A., Van Den Berg, M., Lu, W., Lubbad, R. & Løset, S. (2018), Modelling results with a new simulator for arctic marine structures - SAMS, in 'Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE', Vol. 8, pp. 1–8.
- Walsh, J. E. (2008), 'Climate of the arctic marine environment', *Ecological applications* **18**(sp2), S3–S22.
- Woodyard, D. (2004), Sulzer (Wärtsilä), in 'Pounder's Marine Diesel Engines', Butterworth-Heinemann, pp. 641–663.

# Appendix I: Ice loads

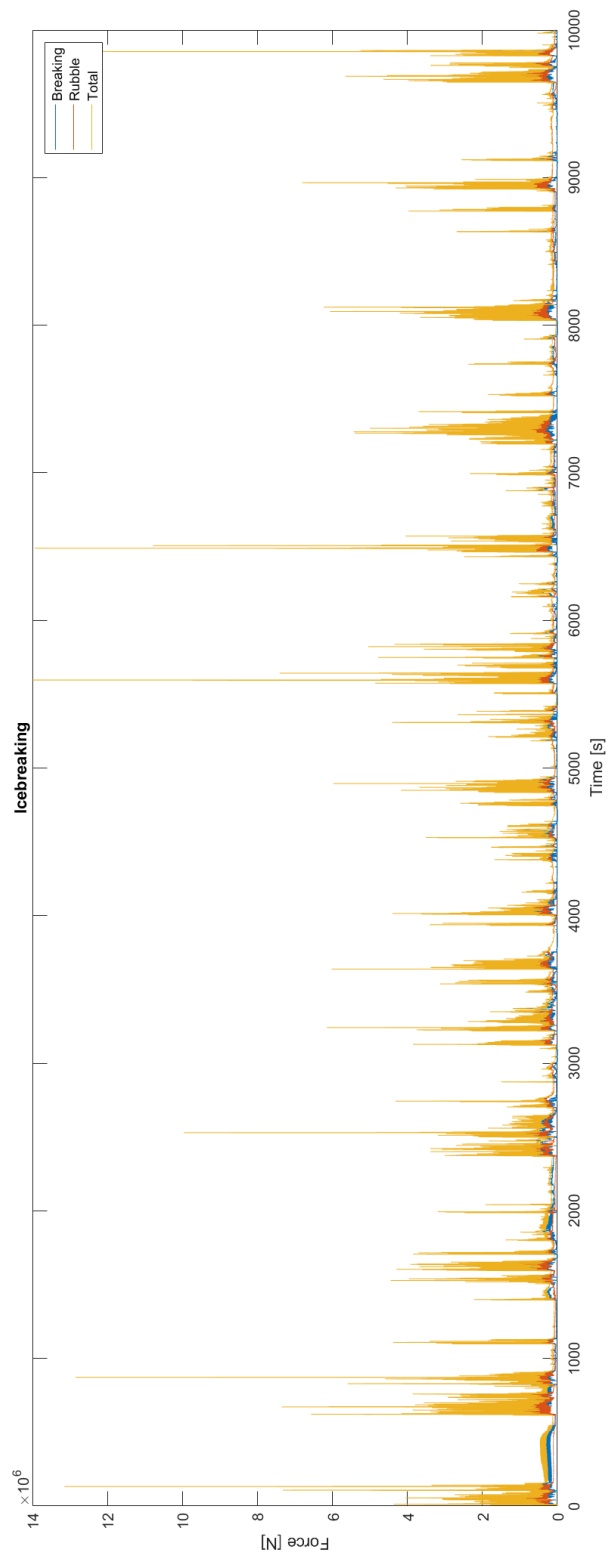


Figure 33: Ice loads

## Appendix II: Systemstructure.ssd

```
<?xml version="1.0" encoding="UTF-8"?>
<ssd:SystemStructureDescription name="simple-cse-example" version="
  Draft06032020">
  <ssd:DefaultExperiment startTime="0.0" stopTime="2e-3"/>
  <ssd:System name="cse-instance" description="An example of how to
    use Core Simulation Environment">

    <ssd:Annotations>
      <ssc:Annotation type="org.open-simulation-platform">
        <osp:SimulationInformation>
          <osp:FixedStepMaster description="
            FixedStepAlgorithm" stepSize="0.1"/>
          </osp:SimulationInformation>
        </ssc:Annotation>
      </ssd:Annotations>

    <ssd:Elements>
      <ssd:Component name="IceLoads" source="fmu-proxy://localhost
        :9090?file=../Ice_loads.fmu"/>
      <ssd:Component name="Propulsion" source="fmu-proxy://localhost
        :9090?file=../Propulsion_system.fmu"/>
      <ssd:Component name="Hull" source="fmu-proxy://localhost:9090?
        file=../Hull_Resistance.fmu"/>
    </ssd:Elements>

    <ssd:Connections>
      <ssd:Connection startElement="IceLoads" startConnector="output"
        endElement="Propulsion" endConnector="IceLoadIn"/>
    </ssd:Connections>

    <ssd:Connections>
      <ssd:Connection startElement="Propulsion" startConnector="
        velocityIce" endElement="IceLoads" endConnector="input"/>
    </ssd:Connections>

    <ssd:Connections>
      <ssd:Connection startElement="Hull" startConnector="output"
        endElement="Propulsion" endConnector="HullResistance"/>
    </ssd:Connections>

    <ssd:Connections>
```

```
<ssd:Connection startElement="Propulsion" startConnector="
    velocityRecitance" endElement="Hull" endConnector="velocity
"/>
</ssd:Connections>

</ssd:System>
</ssd:SystemStructureDescription>
```



## Appendix III: OspSystemstructure.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<OspSystemStructure xmlns="http://opensimulationplatform.com/MSMI/
  OSPSystemStructure">
  <BaseStepSize>0.1</BaseStepSize>
  <Simulators>
    <Simulator source="Ice_loads.fmu" name="IceLoads"/>
    <Simulator source="Propulsion_system.fmu" name="
      Propulsion"/>
    <Simulator source="Hull_Resistance.fmu" name="Hull"/>
  </Simulators>
  <PlugSocketConnections/>
  <BondConnections/>
  <VariableConnections>
    <ScalarConnection>
      <Source endpoint="output" simulator="IceLoads
        "/>
      <Target endpoint="IceLoadIn" simulator="
        Propulsion"/>
    </ScalarConnection>

    <ScalarConnection>
      <Source endpoint="velocityIce" simulator="
        Propulsion"/>
      <Target endpoint="input" simulator="IceLoads
        "/>
    </ScalarConnection>

    <ScalarConnection>
      <Source endpoint="output" simulator="Hull"/>
      <Target endpoint="HullResistance" simulator="
        Propulsion"/>
    </ScalarConnection>

    <ScalarConnection>
      <Source endpoint="velocityResistance"
        simulator="Propulsion"/>
      <Target endpoint="velocity" simulator="Hull
        "/>
    </ScalarConnection>
  </VariableConnections>
</OspSystemStructure>
```

