



Norwegian University of
Science and Technology

Simulation-Based Analysis of Vessel Performance During Sailing

Describing a simulation platform applied in
early stage ship design

Martin Øksdal Bakke

Peter Slinning Tenfjord

Marine Technology

Submission date: June 2017

Supervisor: Stein Ove Erikstad, IMT

Norwegian University of Science and Technology
Department of Marine Technology



NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER THESIS IN MARINE TECHNOLOGY Spring 2017

For stud.techn.

Peter S. Tenfjord and Martin Ø. Bakke

Simulation-Based Analysis of Vessel Performance During Sailing

The demand for more accurate predictions with regards to the performance of ships and ship systems force scientists and engineers to be innovative with how they assess and design future ships and fleets. Stricter IMO regulations and competition among ship design companies in a tough market has sparked an interest in research for better design methodologies in the industry.

Traditionally, ship designs are assessed using calm water conditions and static statistical approaches. Due to the complexity and scale of ship design projects, systems are commonly designed and tested individually. Complete testing of designs in realistic environments represents a leap forward in terms of understanding how design changes affect the performance of ships at an early stage. Virtual testing of ships allows designers to assess system interactions and key performance indicators more closely linked to the operations.

Replicating vessel performance requires a simulation framework with acceptable accuracy both at a component and a system level. The objective is to create a module-based simulation platform with individual modules to handle all aspects of the ship and its operations provides flexibility in terms of testing and a more intuitive analysis procedure.

Tasks

The candidate shall/is recommended to cover the following tasks in the master thesis:

- a. Provide a description of the background to the problem and discuss how the results from this thesis can improve the situation.
- b. Present state of the art for applying simulation as a tool, focusing on ship design.
- c. Describe the procedure for converting knowledge of marine technology theory into simulation models with practical applications. This includes choosing appropriate theoretical methods.
- d. Investigate how hydrodynamic calculation programmes may be used through the simulation platform.
- e. Build a simulation model in Simulink, that focuses on a single vessel sailing between two ports. This model should contain modules for weather, hydrodynamics and machinery.
- f. Verify that the simulation model is working properly, meaning that it does calculations correctly, that the communication between the modules is working, and that data is exchanged correctly and at the right time steps.
- g. Validate the results obtained from the model, i.e. make sure that the model provides output that may be used to understand the real-life system. In order to make sure that the model doesn't have systematic faults, comparisons should be made to real-life data or other, validated models.
- h. A part of the expansion to more complex cases is to decide on experimental design. This means that we have to decide how the simulation runs should be set up, in order to answer design questions. Find out how different cases may be used to answer different design questions, and what parameters that warrant further investigation.



General

In the thesis the candidate shall present his personal contribution to the resolution of a problem within the scope of the thesis work.

Theories and conclusions should be based on a relevant methodological foundation that through mathematical derivations and/or logical reasoning identify the various steps in the deduction.

The candidate should utilize the existing possibilities for obtaining relevant literature.

The thesis should be organized in a rational manner to give a clear statement of assumptions, data, results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Telegraphic language should be avoided.

The thesis shall contain the following elements: A text defining the scope, preface, list of contents, summary, main body of thesis, conclusions with recommendations for further work, list of symbols and acronyms, reference and (optional) appendices. All figures, tables and equations shall be numerated.

The supervisor may require that the candidate, in an early stage of the work, present a written plan for the completion of the work.

The original contribution of the candidate and material taken from other sources shall be clearly defined. Work from other sources shall be properly referenced using an acknowledged referencing system.

The thesis shall be submitted electronically (pdf) in DAIM:

- Signed by the candidate
- The text defining the scope included
- Computer code, input files, videos and other electronic appendages can be uploaded in a zip-file in DAIM. Any electronic appendages shall be listed in the main thesis.

The candidate will receive a printed copy of the thesis.

Supervision:

Main supervisor: Professor Stein Ove Erikstad
Co-supervisor: PhD Candidate Endre Sandvik
Deadline: 11.06.2017

Trondheim, 07.06.2017

Stein Ove Erikstad
Supervisor

Preface

This master thesis is written at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU) in Trondheim during spring 2017. The thesis is the final part of graduating with the degree *Master of Science* (Siv.Ing.) with specialization within Marine System Design.

The thesis has its foundation in the SFI Smart Maritime's work package four, and is aimed at simulation of a ship system with the necessary sub-systems included. It is written in cooperation with Endre Sandvik, who is working as a PhD candidate for the mentioned project.

We would first like to thank our supervisor, Professor Stein Ove Erikstad, for good guidance and useful discussions throughout the semester. Our co-supervisor, PhD candidate Endre Sandvik also deserves thanks for his dedication towards our learning advance in the subject and helpful suggestions. We would also like to thank Dariusz Fathi at SINTEF Ocean for his support with the software Shipx, and SFI partners Grieg Star for providing case data.

Trondheim, June 9, 2017



Peter S. Tenfjord



Martin Ø. Bakke

Summary

Traditionally, a ship design is evaluated based on how it performs in idealized conditions. This approach fails to capture the complete spectrum of real-life operating conditions, as the seas are seldom calm. This thesis presents a simulation based method for evaluation of early stage ship designs. By creating realistic environmental conditions, and evaluating how the vessel performs under these conditions, the thesis aims to improve the accuracy of ship design evaluation.

In this thesis a module based platform combining hydrodynamic calculations, weather data, and simulation of vessel operation, has been developed. In order to capture the influence of environmental conditions, the added resistance in waves and wind was included. The hydrodynamic workbench ShipX, developed by Marintek, was integrated into the platform. An alternative method based on Hollenbach's empirical resistance calculation has also been applied.

The simulation platform was validated in a case study with a 50,000 dwt open hatch carrier. The case vessel was implemented in the platform, and a model of the hull was imported to ShipX. A voyage undertaken by the case vessel was recreated, using hind-cast weather data from points along the route. The platform estimated 16 and 18% larger average break power, using ShipX and Hollenbach respectively, compared to case data. Compared to the model tests for the vessel the calculated calm water resistance was estimated to be roughly 18% more for both methods at 15 knots. Large fluctuations in the estimated break power need was found, indicating that the platform is sensitive to changes in the simulated environmental conditions.

The case was expanded by changing the initial design of the vessel. Three altered designs have been tested in the sailing scenario, with increased breadth, and trim both fore and aft. The results showed a 5.3% increase in average break power when the breadth of the ship was increased by one meter. A 3.4% increase was found when the fore trim was increased by one meter, and 1.5% decrease was found when the fore trim was decreased by one meter. When evaluating and comparing vessel changes, the platform has produced satisfactory results as to which design performs better. It has been shown that the effect one change has on the entire system is easy to assess. This entails the ability to find the best solution for the entire system, which can be different from the best solution for the sub-system.

The platform was able to capture important interactions between a ship and its subsystems, and how the ship performs under realistic weather conditions. It improves the decision making process by providing the designer with a tool for rapid design testing and performance evaluation.

Sammendrag

Den tradisjonelle fremgangsmåten for evaluering av skipsdesign er basert på hvor bra det yter under ideelle forhold. Denne tilnærmingen vil ikke kunne gjengi et komplett bilde av reelle tilstander. Denne oppgaven presenterer en simulerings-basert metode for å kunne evaluere et skipsdesign i tidlig fase. Ved å realistisk gjengi tilstandene i omgivelsene til skipet, og evaluere ytelsen under slike forhold. Oppgaven tilstreber å forbedre presisjonen ved evaluering av skipsdesign.

I denne oppgaven har det blitt utarbeidet en simuleringsplattform som kombinerer hydrodynamiske beregninger, værdata og et skips operasjoner. For å se effekten av vær på et skip har tilleggsmotstand blitt beregnet. ShipX, en hydrodynamisk arbeidsbenk utviklet av Marintek, ble integrert i plattformen. Et alternativ til ShipX ble inkludert ved med Hollenbach's empiriske metode for motstandsberging.

Simuleringsplattformen ble validert gjennom et case studie der et 50 000 dwt bulkskip ble brukt. Skipets egenskaper ble implementert i plattformen, og en modell for skipet ble importert til ShipX. En av skipets sjøreiser ble gjenskapt ved å bruke historiske vær-data fra strategiske punkter langs reisen. Det ble estimert 16% og 18% høyere effektbehov ved bruk av ShipX og Hollenbach, respektivt. For seiling i 15 knop ble stille vannsmotstand beregnet til å være ca. 18% høyere i forhold til modelltester gjort for skipet, dette gjaldt begge metodene. Det ble observert stor varians i effektbehovet, som indikerer at modellen er svært sensitiv til forandringer i været. Skipet fra casen ble brukt videre for å se på effekten av å endre lastkondisjon og dimensjoner. Tre endrede design ble evaluert i samme sjøreise som tidligere. Ett med endret bredde, mens de to andre fikk endret trim. I forhold til skipet i design tilstand, økte effektbehovet med 5,3% for skipet med økt bredde. For skipet med baugen senket en meter, minket behovet med 1,5%. Skipet med en meter hevet baug viste et økt behov på 3,4%.

Under evaluering og sammenligning av endringer på skrog eller lastkondisjon har plattformen gitt tilfredsstillende resultater, der man kan se hvilket design som yter best. Det har blitt vist at endringer og deres påvirkning på systemet enkelt kan finnes. Dette medfører at man kan søke etter den beste løsningen for systemet som helhet, som kan være ulikt det individuelle systemets beste løsning.

Plattformen er i stand til å gjengi viktige sammenhenger mellom skipet og dets undersystemer, og hvordan skipet yter under realistiske vær-forhold. Den forbedrer beslutningsprosessen ved å gi designeren et verktøy for hurtig design testing og evaluering av ytelse.

Contents

Preface	i
Summary	iii
Sammendrag	v
Nomenclature	xiii
1 Introduction	1
1.1 Background	2
1.2 Objective	3
1.3 Previous Work	4
1.4 Limitations	4
1.5 Thesis Structure	5
2 Simulation as a Tool in Ship Design	7
2.1 Ship Design	7
2.2 Simulation	11
2.3 Simulation in Product Development	14
2.4 Current Use of Simulation	14
2.5 Introducing Simulation of Maritime Systems and Operations	15
2.6 Numerical Ship Simulators	17
3 Software	23
3.1 MATLAB	23
3.1.1 Simulink	23
3.1.2 SimEvents	24
3.1.3 Batch Mode	24
3.2 ShipX	24
3.2.1 ShipX Vessel Responses (Veres)	25
3.2.2 ShipX Wave Resistance (Waveres)	27
4 Problem Description and Simulation Platform	29
4.1 Problem Description	29
4.2 Introduction of the Platform	30

4.2.1	Purpose	33
4.2.2	Rule Based Decision Making	34
4.2.3	Desired Output and Expected Interpretation	34
4.3	Scenarios for Simulation	35
4.3.1	Sailing	35
4.3.2	Round Trip	35
4.3.3	Heading and Route	36
5	Simulation Platform and Modules	39
5.1	Operation Module - Simulation of Vessel Operation	39
5.1.1	SimEvents Sailing Model	40
5.1.2	SimEvents Round Trip Model	41
5.2	MetOcean Module - Simulating Operating Conditions	42
5.2.1	Stochastic Approach to Modeling Weather Conditions	44
5.2.2	Random Number Generation	45
5.2.3	Wave Spectrum	45
5.2.4	Processing Weather Data	46
5.3	Hydrodynamics Module - Ship Resistance	48
5.3.1	Ship Resistance	49
5.3.2	Resistance Prediction	51
5.3.3	Hollenbach Model	52
5.3.4	ShipX Model - Veres and Waveres	52
5.3.5	Speed Loss Model	54
5.4	Machinery Module - Estimation of Total Fuel Consumption	56
5.4.1	Propulsive Power	57
5.4.2	Engine Model	58
5.4.3	Engine Model With Energy Storage System	59
5.5	Assembling the Modules	60
6	Results and Validation	63
6.1	Case: Star Lysefjord	63
6.1.1	Case Description	63
6.1.2	Results	65
6.2	Altering trim and dimensions	70
6.3	Varying Speed	73

6.4	Simulating a Ship with Energy Storage System	75
7	Discussion	77
7.1	Method and Application	77
7.2	Resistance Calculation	77
7.3	Computational Time	79
7.4	Modeling MetOcean Conditions	80
7.5	Estimating Power Requirement	81
7.5.1	Power as a Function of Speed and Resistance	81
7.5.2	Auxiliary and Hotel Loads	82
7.5.3	Hull Fouling and Degradation	82
7.6	Accuracy of the Model	83
8	Conclusion and Further Work	85
8.1	Conclusion	85
8.2	Further Work	86
	Bibliography	89
A	Electronic Appendix	92
B	Coding	93
B.1	Main Script used to run Simulation	93
B.2	Route Interaction	99
B.2.1	Route Generation	99
B.2.2	Determining position of vessel at time, h	101
B.3	Markov chain	102
B.3.1	Generation of Markov chains	102
B.3.2	Next state is found	106
B.4	Hollenbach	109
B.5	Interaction with ShipX - Veres	112
B.5.1	Calculating added resistance	112
B.5.2	.bat file to initiate Veres through batch mode with the correct input	115
B.6	Organization of vessel parameters	116
C	Data Sheets from ShipX	117

List of Figures

2.1	The ship design spiral (MarineWiki, 2011)	8
2.2	Concept of required EEDI, reduction factor, cut off limits and EEDI phases (IMO, 2016)	10
2.3	Illustration of problem solving methods and when each method should be used	13
2.4	Illustration of the Functional Mockup Interface (Blockwitz et al., 2012)	15
2.5	An overview of the total ship simulator by Bassam et al. (2016)	19
3.1	Input vs. calculated values from Veres (Fathi and Hoff, 2017)	26
4.1	Overview of simulation platform	31
4.2	Illustration of great circle and rhumb lines, between Tokyo and L.A.	37
5.1	SimEvents sailing model	41
5.2	SimEvents round trip model	42
5.3	MetOcean module	43
5.4	Empirical prob. dist. for peak wave period during head seas, mid-pacific (BMT ARGOSS, 2017)	47
5.5	Hydrodynamics module	48
5.6	Subdivision of total resistance into components, for a vessel sailing in calm seas. Based on Amdahl et al. (2005)	49
5.7	A typical resistance curve showing the composition of the total resistance coefficient for different Froude numbers for calm seas	50
5.8	Comparison of post-processors for $H_s = 4\text{m}$ and $\beta = 0^\circ$	54
5.9	Machinery module	57
5.10	Engine model with one main engine	58
5.11	Engine model with one main engine and a battery	59
5.12	Hydrodynamic and machinery modules assembled in the ship model	60
5.13	Top level view of MetOcean and operations module, and ship model	61
6.1	Plot of the voyage from Yancheng to Panama. The boxes show where MetOcean data is collected	64
6.2	Comparison of the break powering need estimated by simulation, and the operational data	66

6.3	Percentage deviation in break power between data set, Hollenbach and ShipX . . .	66
6.4	Encountered combined wave height for the Hollenbach run	67
6.5	Encountered wave direction for the Hollenbach run	67
6.6	Encountered wave heights, sea and swell, for the ShipX run	68
6.7	Encountered wave direction, sea and swell, for the ShipX run	68
6.8	Resistance components and total resistance for Vessel 2 during transit	71
6.9	Empirical probability distribution for significant wave height period during transit	72
6.10	Empirical probability distribution for zero-crossing wave period during transit . .	72
6.11	Attainable speed with relevant wave height and direction for Vessel 2 during transit	73
6.12	Resistance components for Vessel 2 during transit	74
6.13	Break power for Vessel 2 during transit	74
6.14	Power demand and battery charge level during transit	75
7.1	Calm water resistance for case vessel at design conditions	78
7.2	Percentage deviation from trial results in figure 7.1	78
7.3	Correlation between weather parameters	81
C.1	Case condition	117
C.2	Design condition	118
C.3	Vessel 1	119
C.4	Vessel 2	120
C.5	Vessel 3	121

List of Tables

4.1	Overview of the components in the simulation platform	29
5.1	Description of the attributes of the ship entity	40
5.2	Output from the MetOcean module	43
5.3	Markov chain transition matrix for significant wave height in the Pacific	44
5.4	Values corresponding to the respective states found in table 5.3	45
5.5	Parameters available from Waveclimate	47
5.6	Parameters for Kwon's method, formula 5.6 and 5.7	56
5.7	Speed reduction coefficient C_U by block coefficient (Lu et al., 2015)	56
5.8	Direction reduction coefficient C_β due to weather direction (Lu et al., 2015)	56
5.9	Ship form coefficient C_{form} due to ship categories and loading condition (Lu et al., 2015)	57
5.10	Efficiencies and their typical values for ships with one propeller(MAN B&W Diesel, 2011; Bertram, 2012)	58
6.1	Star Lysefjord particulars (Grieg Star, 2013)	63
6.2	Description of data from Star Lysefjord	64
6.3	Data from case voyage	64
6.4	Summary of results from case voyage simulation	69
6.5	Effects that a change of dimension have on the results	71
B.1	Location of parameters in the excel spreadsheet: Vessel info	116

Nomenclature

Acronyms

BN	Beaufort Number
CFD	Computational Fluid Dynamics
DES	Discrete Event Simulation
EEDI	Energy Efficiency Design Index
EEOI	Energy Efficiency Operational Indicator
FMI	Functional mock-up interface
IMO	International Maritime Organization
IMR	Inspection, Maintenance and Repair
JONSWAP	Joint North Sea Wave Project
KPI	Key Performance Indicators
LCC	Life Cycle Cost
MCR	Maximum Continuous Rating
MDO	Marine Diesel Oil
ME	Main Engine
OSV	Offshore Service Vessel
PSD	Preliminary ship design
PSV	Platform supply vessel
SFI	Centres for Research-based Innovation

Greek Symbols

β	Relative direction of oncoming weather	degrees
η_D	Propulsive efficiency	-
η_H	Hull efficiency	-

η_M	Mechanical efficiency	-
η_O	Propeller open water efficiency	-
η_R	Relative rotational efficiency	-
η_S	Shaft efficiency	-
η_T	Total efficiency	-
ρ	Density of sea water	kg/m^3
ρ_{air}	Density of air	kg/m^3
ζ_W	Wave amplitude	m
Other Symbols		
A_p	Transverse projected area of ship superstructure	m^2
C_B	Block coefficient	-
C_F	Amount of CO_2 generated per mass of fuel used	-
C_U	Speed reduction coefficient	-
C_β	Direction reduction coefficient	-
C_{Form}	Form reduction coefficient	-
F_{ds}	Wetted surface correction factor	
P_B	Break power	kW
P_D	Delivered power	kW
P_E	Effective power	kW
P_T	Thrust power	kW
R_N	Reynold's number	-
k	Form factor	-
S	Wetted surface	m^2
SFC	Specific fuel consumption	g/kWh
V	Ship speed	m/s or knots

1. Introduction

Recent changes in the offshore industry has lead to more competition amongst the actors, which in turn has resulted in an increased focus on cost savings. Since the design process ahead of procurement can account for up to one third of the total lead time in a shipbuilding process, there is great potential for improvement both with respect to lead-time and to costs. The ever-changing need for new designs and solutions does not make it any easier, as it limits the possibilities for re-use.

Another trend the shipping industry has experienced in recent years is the increased focus on green ships, with ship designers scrambling to improve the energy efficiency of their designs. Improved hull geometry and other innovative design solutions are in high demand as the IMO and other actors tighten the emission regulations. With so much of the world trade going by sea, energy efficient ships are almost a prerequisite if we are to reach the reduction of green house gas emissions that the international community has agreed upon.

Ship owners and operators also want to reduce fuel consumption, although their desire has a more monetary drive. Fuel costs typically account for large parts of a ships operational expenses, and in a tough market where cost margins are being tested, reducing fuel costs is key to maintaining competitiveness.

Energy efficiency can be improved in different ways, from optimizing designs to weather routing of ships and other operational measures like speed management. In order to maximize the effect of these actions it is important to adjust them according to the operating profile of the ship. Wrongful estimation of the conditions in which the ship operates leads to a design that is optimized for other conditions than the actual. This may in turn cause reduced performance, and can have large economical consequences.

This thesis investigates how simulation can be used to virtually test designs early in the design process. This is to be done by establishing a simulation framework that combines weather data with hydrodynamic calculations and ship operations, thus providing the designer with more accurate estimates towards a ships performance. The end goal is that the simulation model can assist in a decrease of lead-time and costs during a design process while the end product is improved.

1.1 Background

Ship design has long been somewhat of an art-form, where experience, know-how, tradition and trial-and-error has been the name of the game. This is not because ship designers are slow-movers and late to take advantage of modern computer based aids. More likely it is because ships are so complex and intricately connected, and more analytic approaches have not improved the performance of the designs compared to the old ways.

The demand for more accurate predictions with regard to the performance of ships and ship systems force scientists and engineers to be innovative with how they assess and design future ships and fleets. Stricter IMO regulations and competition among ship design companies in a tough market has sparked an interest in research for better design methodologies in the industry. Recent projects like VISTA ([Erikstad et al., 2015](#)) and ViProMa ([Hassani et al., 2016b](#)) have paved the way for the work in this thesis.

A vital part of ship design is to understand how, and under what conditions, the ship will operate. The accurate prediction of an operational profile is important, as it represents the conditions for which the ship will be optimized. Traditional methods for evaluating designs do not offer the opportunity to handle complex profiles, and normally make simplified calculations based on an average or representative condition ([Fathi et al., 2013](#)).

[Gaspar et al. \(2012\)](#) has investigated how complexity is handled in conceptual ship design. Their work shows that there is a constant increase in the amount of information required to design a ship, and it is necessary to have methods able to handle this new information and to bring other type of insights during the preliminary stage. This is not just technical and economical information, but also environmental performance and risk. One way to rapidly and inexpensively provide and process large amounts of information is through simulation.

Simulation

Simulation can be described as imitating the operation of a real-world system or process over time. This generates an artificial history to draw inferences concerning the operating characteristics of the real system that is represented ([Banks et al., 2000](#)). In other words, simulation is a means of analysing a real system, in order to predict how this real system will perform.

Simulation is an example of a method that can be used to evaluate ship designs. In a simu-

lation model large amounts of information can be added, without reducing the usability and understanding of the system to be modeled. Large time-scales can be simulated rapidly, and results presented in a graphical way that enables the user to see clearly the performance of the system.

Due to the complexity and scale of ship design projects, systems are commonly designed and tested individually. Complete testing of designs in realistic conditions represents a leap forward in terms of understanding how design changes affect the performance of ships at an early stage. Virtual testing of ships can allow designers to assess system performance and how it is linked to the operating conditions in a dynamically changing environment. If the operating conditions and boundary conditions can be represented accurately, vessels can be compared in true-to-life conditions (Fathi et al., 2013).

Replicating vessel performance requires a simulation framework with acceptable accuracy both at a component and a system level. Creating a module-based simulation platform with individual modules to handle all aspects of the ship and its operations provides flexibility in terms of testing and a more intuitive analysis procedure. Such a platform would allow for benchmarking of designs, where the criteria for evaluation can be a number of key performance indicators.

1.2 Objective

The aim for this thesis is to build a simulation platform making it possible to simulate a vessel during sailing, by virtually creating an operating environment. The purpose of this platform is to estimate the performance of the vessel in a more accurate way than is currently done by designers, at an early stage in the design process. This is achieved by building a platform that requires relatively little input to quickly produce performance indicators based on simulated operating profiles. In addition the platform should require limited configuration time from the user. To facilitate this, a robust and flexible framework should be in place. Only ship and mission related input is required, and once this information is entered the platform should be ready to run. The platform should have a modular design, where the modules are interchangeable depending on requests towards fidelity level and operation type.

The platform will benefit the designer as it provides rapid feedback on the design performance already in the early design phase. The platform should allow design changes to be made and evaluated, with little costs attached. As more detail is added to the ship and its mission, the

platform can produce more detailed and realistic operating profiles and performance indicators. Lifetime simulations will allow the designer to showcase the design, and compare it to others. Hence the platform can also be used as a decision support tool by owners and operators when choosing between different concepts, based on their weighting of the performance indicators.

The objective of this thesis can be summarized by the following research questions:

- Can a simulation platform be built that is able to replicate real-life operations and evaluate ship performance with realistic output?
- How can the ship-ocean system be described, and what theoretical methods are required?
- Can the simulation platform improve the final design while also reducing the lead-time?

1.3 Previous Work

In the project thesis an attempt was made to describe a platform capable of simulating the operation of a vessel over its intended lifetime. The project gave good insight into how such a model should be developed, and made up a foundation for this master thesis. The project thesis also made it clear that developing a complete simulation platform for all types of vessels and operations is an extensive process. The project thesis is used to choose in which direction the scope of the master thesis should move, and how it should be limited.

1.4 Limitations

The scope of this thesis and its objectives have the following limitations

- It is not a goal for this thesis to build modules adapted to all existing ship types and operations, but merely to show how this can be possible through a couple of modules that are interchangeable.
- Only displacement type ships are included in this work.
- We will limit ourselves to look at one vessel at the time, and not fleet operation.
- Only the propulsive power need is included, meaning that hotel and other auxiliary power consumers are left out.

- There was only available data from one case and vessel, limiting the possibilities for validation.

1.5 Thesis Structure

This thesis will first provide separate introductions to ship design and then simulation. The simulation part will have a general introduction first, with a consecutive view on how simulation is applied in the maritime industry today.

In chapter three the software used in this thesis is presented along with their relevant areas of application.

Chapter four presents the problem description for this thesis and an introduction to the idea of a simulation platform, the purpose of it, and expectations to its content and final results.

Chapter five presents the individual modules included in the platform. The theoretical approach chosen and the thought behind every module is explained, as well as how the modules interact.

In chapter six the vessel Star Lysefjord of Grieg Star is used as a case to validate the simulation model. With an available 3D model, and sailing data, it was possible to replicate one of Star Lysefjord's voyages. Further on results from the simulation model not relevant for the specific case of Star Lysefjord are presented. These results will reflect the impact design changes have on the resistance for the vessel, as well as the affect of trim change.

The last two chapters are discussion, and conclusion and recommendations for further work.

2. Simulation as a Tool in Ship Design

This chapter will start with an introduction to the traditional ship design process, and factors affecting decision making during the design of a ship. Then the reader will get an introduction to the field of simulation, first with the basic terms used to describe computer simulations. We then look at simulation tools in operations research, and describe some aspects of using the simulation approach. Finally, a review of relevant literature is presented, where the state of the art within ship simulation models is investigated.

2.1 Ship Design

Ship design is a challenging task that span over many disciplines, from hull design to machinery and structural engineering. It involves numerous engineering fields, from hydrodynamics to mechanics and thermodynamics, and of course economics. A characteristic of ship design is that you have to make assumptions about the final design in order to evaluate the current solution. These assumptions in turn have to be tested, and the design modified according to the results.

The ship design process is often described as a spiral, following the early work of [Evans \(1959\)](#). A more modern take on this spiral can be seen in figure 2.1. Along the points on the curve of the spiral, information is added, details are drawn and engineering calculations are done. As the work proceeds, the spiral is repeated until the design is finished. The basic idea is that the design process is sequential and iterative, rather than concurrent.

Ship design is commonly divided into phases, or stages, where each phase has an objective and a set of tasks to achieve. Ship designers might have different takes on how the design process is described. In the following, a typical description is used. In figure 2.1 the phases are separated by how close they are to the centre of the design spiral - the final design. The first phase is called the concept design phase, and the key objective of this phase is to investigate the feasibility of the project at hand. In this phase, more than one design might be developed. These concepts must then be evaluated against each other, based on how well they accomplish the mission requirements. In the preliminary design phase the project is planned, and the key objective is to try to prove that the selected concept should move into the contract design phase. In this phase, the objective is to arrive at the specifications, providing a costing of the project. Finally, the de-

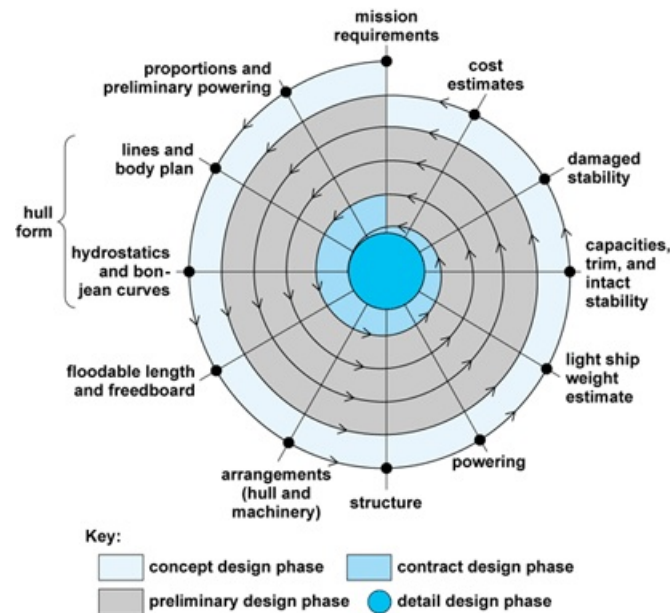


Figure 2.1: The ship design spiral (MarineWiki, 2011)

tail design phase produces drawings for the ship builder, and documentation from classification societies etc. This ends the design spiral.

It is also common to describe the ship as a system, and development in systems engineering has influenced the ship design process (Erikstad, 2015). However, the complexity of the ship and its subsystems poses challenges for the designers, and for most methods an iterative process of mapping the connection between design and performance is used. One of the challenges, as described by Erikstad et al. (2015), is that everything is connected, and a change in one component can have large impacts on other subsystems and the overall performance of the ship.

The ship designer must always make compromises, facilitating the needs of the ship owner into the design. For example, the breadth of the ship might be increased to increase cargo capacity, but that will also result in increased resistance and subsequently power demand. Often, the final decision boils down to the economics - if the revenue from the increased cargo capacity exceeds the increase in building and life cycle costs (LCC), the breadth should be increased. However, the effects of changing the breadth might not be clear-cut, and it will very much depend on the operating profile of the ship, and market developments.

Operating Profile

An operating profile, or operational profile, aims to predict how a ship will operate over a period of time. It describes under which weather and loading conditions the ship will sail. An operating profile typically contains the percentage of time spent in port, and sailing, both laden and in ballast. It can also include a speed distribution, draft ranges, time spent manoeuvring, time spent in DP, and machinery utilization. The contents of the operating profile will also differ for various ship types performing different types of operations, from the straight forward deep sea shipping, to more complex offshore operations.

A ship design is typically carried out by optimizing the hull form for a limited range of idealized operating conditions. These so called design conditions can be found by analyzing the operating profile to find the conditions in which the vessel should operate most of the time. However, in reality a vessel only operates in its design condition a small proportion of the time (Banks et al., 2013). This means that the designer does not take into account the full spectrum of different operating conditions, weather and sea states, or logistics requirements, i.e. real world factors that will alter the operations (Erikstad et al., 2015).

Consequently, there might be large deviations between the real life operating conditions and how the designer imagined the ship to operate. This might lead to differences in fuel consumption and sailing time, which can cause economical consequences. Therefore it is important to be able to attain a realistic operating profile, so that the difference between estimated design performance and actual operation performance can be minimized.

Design Performance

A common problem for the ship designer is to demonstrate the effect of potential design changes before vessels are built and brought into operation (Fathi et al., 2013). Most shipowners focus on the economical performance of the ship, return on investment and low OPEX and CAPEX. In recent years there has also been an increased attention on the environmental performance of the ship, which is often closely related to the OPEX through energy efficiency, because of the relatively high proportion of costs going to fuel.

It can also be a challenge to decide on what parameters the performance of the ship should be measured. Energy efficiency, i.e. fuel consumption has always been important. Recent years have also seen the shipping industry focus on greener designs, both to reduce costs, and to take

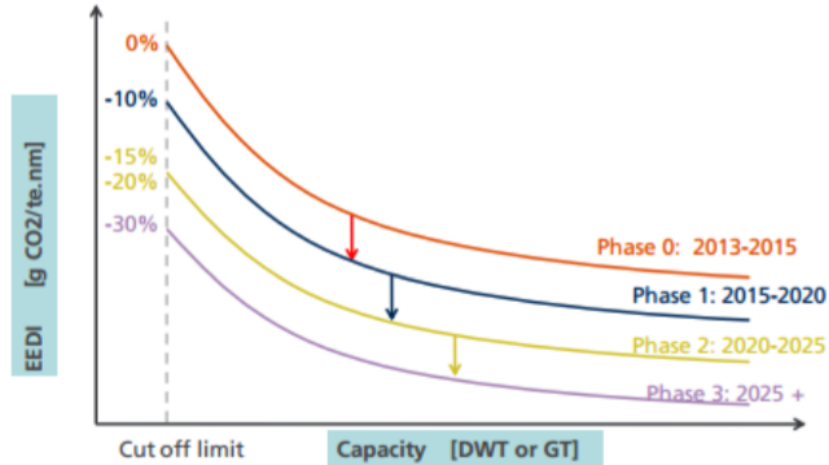


Figure 2.2: Concept of required EEDI, reduction factor, cut off limits and EEDI phases (IMO, 2016)

action against global warming. Other factors that can be used to evaluate design performance are attainable speed and ship motions in seas, DP performance, payload capacity, and more. In order to assess the overall performance, the designer will have to weigh the importance of the chosen factors, often according to input from the owner.

The International Maritime Organization, IMO, has introduced maritime energy efficiency regulations in order to benchmark the energy efficiency of new ship designs (IMO, 2016). These regulations include amongst others the Energy Efficiency Design Index, EEDI. The EEDI is an index that indicates the energy efficiency of a ship in terms of how much CO_2 it emits per cargo transported, as shown in the simplified formula below.

$$EEDI = \frac{CO_2 \text{ Emission}}{\text{Transport Work}} = \frac{\text{Engine Power} * SFC * C_F}{Dwt * Speed} \quad (2.1)$$

where SFC is the specific fuel consumption and C_F specifies the amount of CO_2 generated per unit mass of fuel used. There are also more factors included in order to account for the differences in ship types and operation. The EEDI benchmarks the design of a new ship against a reference line giving an allowable EEDI value limit for a given deadweight (IMO, 2016).

Figure 2.2 shows the reference line becoming more stringent over time. The IMO states that the intention behind limits on the EEDI is to drive ship technologies to become more energy efficient over time. As all new ships with a gross tonnage of more than 400 tonnes have to comply with these new regulations, the IMO aims to make quite an impact on the shipping industry. How a design does on the EEDI is therefore an important performance indicator, and a good

example of how the performance of a design could be evaluated.

Sea Margin

Ship design is often based on experience, and perhaps nowhere is this seen more clearly than with sea margin, or service margin. The sea margin incorporates the increased resistance due to wind, waves and currents. It also accounts for hull and propeller degradation, which will happen over time. The purpose of the sea margin is for the ship to be able to keep its design speed also after the resistance increases. Typical values for the sea margin ranges between 15 and 30% of the required engine power, depending on the ships intended area of operation (Amdahl et al., 2005). The sea margin is commonly multiplied with the estimated calm water powering need, to obtain the installed power need of the propulsive machinery. Clearly, the inclusion of a sea margin has a huge effect on the fuel consumption and cost of the ship, both CAPEX and OPEX.

2.2 Simulation

Basic Terms of Simulation

Simulation is widely used to analyse stochastic systems. A stochastic system is a system that evolves probabilistically over time, and is unpredictable because of one or more random variables (Sánchez, 2007). In the following, some of the terminology used in simulations is described, following Sánchez (2007). These terms form the basic concepts of simulations. They will be repeated throughout the text with specific explanations where it is deemed necessary.

A *system* is a collection of entities that interact. An *entity* is a component in the system that requires explicit representation. This can be for example a ship or a cargo, and they have different attributes. *Attributes* are the properties of a given entity, and can for instance be the capacity of a ship, or the size of a cargo.

The *system state* is a collection of variables needed to describe the system at any time, i.e. the state of the system can be completely determined by this set of variables. The variables whose values define the state of the system are called *state variables*. An example of a state variable is the length of a queue or amount of cargo available.

A change in the system state is called an *event*. Examples of events are arrivals of new tasks, initiation of a task, or the departure of a task. A task can for example be the loading of a ship. Events can be endogenous or exogenous, meaning that they can originate from within the system itself, or as an effect of an external cause, respectively.

A *model* is an abstract representation of a system, usually containing structural, logical or mathematical relationships that describe a system in terms of state, entities, events and so on. This means that the model keeps track of how the variables are influenced by an event, and how this in turn affects the system state.

Models can vary a lot in terms of how they are defined, and they can be placed into different categories based on their characteristics. A model in which the system state is defined at all times is called a *continuous-time model*. If the system state is defined only at particular instants in time, the model is called a *discrete-time model*. Also the state of the models can be continuous or discrete. The model is called a continuous- or discrete-state model depending on whether the state variables are continuous or discrete. A discrete state model is also called a *discrete event model* (DES), and similarly for a continuous event model.

A model can be either *deterministic* or *probabilistic*. To be deterministic means that one can predict the results of the model with certainty before running it. A probabilistic model gives different results every time it is run, for the same set of input parameters.

Whether or not time is a variable also divides models. A model is called *static* if time is not a variable. If the system state changes with time, the model is called *dynamic*. The model can also be linear if the output is a linear function of the input, and nonlinear otherwise.

In general, most computer simulation models are continuous-time, discrete-state, probabilistic, dynamic and nonlinear (Sánchez, 2007).

Types of Simulation

Simulation appears in different forms, spanning from high-level transport systems to detailed models of cars and other products. Different projects use simulation for various purposes, and the use of a simulation approach will have to be justified in each case. Sometimes you might be better off by choosing different approaches, like direct calculations or a statistical analysis. By better off we mean that you may perhaps more rapidly arrive at a similar result, or avoid errors by not including elements required by a simulation.

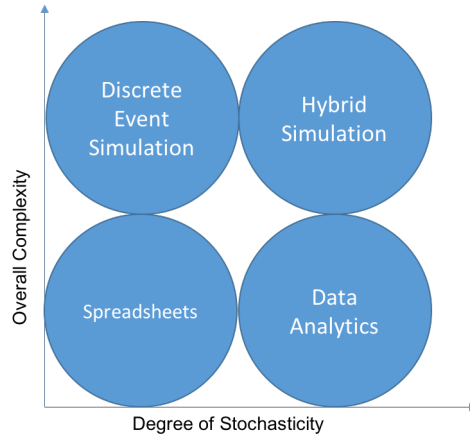


Figure 2.3: Illustration of problem solving methods and when each method should be used

We may use figure 2.3 to illustrate some different problem solving approaches, and show when the use of simulation could be beneficial. The degree of stochasticity is meant to illustrate to what degree you may accurately predict the outcome of the calculations. If you know all the variables, it could be smarter to use spreadsheet simulation than to set up a discrete event model. A spreadsheet simulation, for example in excel, can be sufficient also for more uncertain systems. However, it can quickly become overwhelming, and the interface might not be suited once the number of different subsystems increases, i.e. increasing complexity.

If the complexity is quite low, and there are sufficient data sets available, one might analyze the data directly. The field that is data analytics might also involve computer programs. As the overall complexity increases, more specialized simulation software becomes interesting. They often have the advantage of being able to handle multiple subsystems in a more graphical and understandable manner. In discrete event simulation software like SimEvents, models are built up of blocks, and one may follow the flow in the system by connecting the blocks. SimEvents may also incorporate uncertainty, but if you want to combine the events with continuous time applications, you may end up with a hybrid simulation model.

In a hybrid model, you can combine both discrete and continuous events, into the same model. This enables you to include more accurate representation of events like filling a fuel tank, loading a tanker, or running an engine.

Considering the above discussion, it seems clear that simulating a vessel during sailing should be done using hybrid simulation. The overall complexity of a ship is quite high, with multiple subsystems and interfaces. The degree of stochasticity is also high, especially when including changing weather conditions.

2.3 Simulation in Product Development

Simulation tools can play an important role in new product development, and have been found to cut development time and costs (Thomke, 1998). A study by Becker et al. (2005), investigated the impact of simulation tools on problem-solving and new product development. They found that simulation tools not only reduce costs and development time, but also improve design quality via the availability of information early in the development process. Through simulation you are able to complete numerous iterations, and isolate the effect of one parameter for each iteration. This enables the product developers to investigate the causal relationship between design characteristics and performance, at a reasonable cost compared to physical prototyping (Becker et al., 2005).

The study by Becker et al. (2005) also found that virtual simulation tools introduced a shift in problem solving strategy. They argue that simulation tools provide the possibility of more innovative designs, under the condition that the possibilities they provide are matched with organizational and management structures required to realize these possibilities (Becker et al., 2005). In other words, to unlock the full potential of simulations a certain work process should be followed.

2.4 Current Use of Simulation

Simulation is currently being used in the maritime industry for various purposes. There exists simulation programs that deals with hydrodynamics, structural analysis, control systems, and fleet operation. In terms of hydrodynamics, and especially maneuvering, there exists models that can accurately simulate the behaviour of the vessel in a given condition. Machinery and power plant simulation has also been the topic of research for recent projects, for instance in Taskar et al. (2016); Bø et al. (2015). For a given power demand, or required speed, they are able to accurately model the responses in the machinery. This gives you not only an accurate estimate of the fuel consumption, but also the utilization of the machinery, and might also help reduce the maintenance and repair cost through optimization of the propulsive configuration.

A problem that arises is that most of these engineering disciplines use their own set of tools, that are rarely inter-operational (Sadjina et al., 2017). This complicates system level simulation,

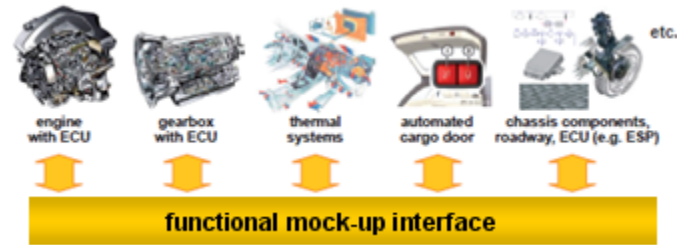


Figure 2.4: Illustration of the Functional Mockup Interface (Blockwitz et al., 2012)

analysis and verification significantly, and may also hide crucial system errors during the design phase. Traditionally, this has led to system simulation models being built from the ground, where the interfaces are customized to such a degree that it is too specific to be re-usable (Sadjina et al., 2017).

Developments in the Functional Mockup Units, and Interfaces (FMU, FMI) may eliminate some of the difficulties in communicating between different system and subsystem models (Blockwitz et al., 2012). The FMI defines a standardized interface to be used in computer simulations to develop complex cyber-physical systems. As illustrated in figure 2.4, the functional mockup interface makes it possible for the different units to communicate. This also enables co-simulation, in which different simulation models are run together.

The idea behind the FMI system is to create a virtual product by combining a set of models that each represent physical parts (Blockwitz et al., 2012). This logic can also be applied towards our purpose, where we try to recreate the actual operation of a ship, by building a virtual ship of different modules, and applying external influences.

The basic idea behind co-simulation is the construction of systems from loosely coupled stand-alone models and the simulation across different subsystems. Co-simulation facilitates the independent exchange and modification of components, and the use of the most suitable tools and solvers for any given subsystem. This also extends to the possibility of separately taking care of initialization, pre-processing, time integration, and post-processing with different specialized tools (Sadjina et al., 2017).

2.5 Introducing Simulation of Maritime Systems and Operations

In the VISTA paper, Virtual sea trial by simulating complex marine operations, (Erikstad et al., 2015) the goal was to assess a vessels performance over its lifecycle, with focus on energy effi-

ciency, improved operability and higher safety level. The paper introduces several challenges, one of which is the challenge of looking at one components influence on the system. Most of a vessels components have complex interactions with each other, and changing one will in most cases influence other components as well, making it difficult to assess the exact effects. This also makes the comparison between vessels difficult, as improvements can be hard to attribute to a specific design choice.

As a successor to the VISTA paper, the ViProMa paper is introduced ([Hassani et al., 2016a](#)). Here they introduce issues regarding virtual prototyping in maritime systems and operations. An issue discussed regarding simulation of such a complex system is the availability of models. Many sub-suppliers will have modelled their equipment using a simulation tool fitting their purpose. This can lead to issues both with respect to compatability between models and sub-suppliers need for discretion. ViProMa's solution to this is black box¹ modelling and a virtual prototyping framework (VPF).

During development of a virtual prototyping framework for maritime systems and operations there are three important factors to keep in mind. First of all the model should make it possible to compare concepts. This requires relevant performance indicators, for example fuel consumption. It would also be of great interest to be able to assess a vessels operability due to weather changes, using MetOcean data². The ability to change main particulars as well as specific operations equipment should also be possible. [Sadjina et al. \(2017\)](#) suggests, as [Hassani et al. \(2016a\)](#) does, to use black box modeling and co-simulation. The idea is to construct a system from stand-alone models, so that the simulation can run across different subsystems. This ensures that each subsystem can be modeled with the most appropriate software.

As the black box environment accounts for a sub-suppliers wish for discretion and confidentiality regarding their products it is a good solution. The drawbacks of such a simulation model are amongst others the lack of transparency, the user of the model will not be able to verify the results in the same way as they would with an open model. Another risk is that suppliers of said models to the platform would see it as an opportunity to showcase their products, making the model perform better than it would in real life. For example by using shop test values for engines where the engine performs under ideal conditions. This would lead to a lower fuel consumption, and one engine used could prove to outshine others in the model while it in real life would perform worse than the others. Another drawback is that the platform can not be fully understood by the users, as the interaction between the modules would be rather superficial and not

¹A black box indicates that the workings of a model are unknown, while the output and input are.

²MetOcean is an abbreviation of Meteorology and Oceanography describing the physical environment at sea.

necessarily have a self-explanatory use, not showing how input is used within each model.

A tool for simulation in the maritime industry that is currently being validated through various cases in cooperation with Wilh. Wilhelmsen and Grieg Star is MARINTEK's GYMir. The GYMir application is built to simulate an integrated ship system. It is said to be an early-stage design assessment that will simulate the long-term performance of a ship in realistic operation profiles. They aim to integrate models of a ship's sub-systems; Hull, Propulsion and Power Systems (Smart Maritime, 2017).

2.6 Numerical Ship Simulators

In this section we review some of the ship simulation models that have been published. We have identified three papers that describe their models and some of the underlying theory, which will be presented in the following. Subsequently, this section will show some alternative ways in which a ship can be modeled for simulation purposes. The purpose of this is to identify the best-practice when it comes to representing the ship, and how to integrate the ship system with the external influences from weather.

Real-Time Marine Vessel and Power Plant Simulation

We start by presenting a conference paper called *Real-Time Marine Vessel and Power Plant Simulation* Bø et al. (2015). Here they present a top-to-bottom overview of the marine power plant and propulsion system, along with a use case of the model on a semi-submersible rig at DP. The model is able to simulate the complex interaction effects between the power plant, hydrodynamics of the propeller (thrusters), and vessel motion. This is achieved by a modular, object-oriented modeling structure, which also allows for different designs and operations to be modeled in the same framework. This means that each block in the simulator represents a physical component in the vessel. Each of these blocks may then in turn consist of various subsystem blocks representing internal physical components of the parent component.

The paper also identifies some use cases for their simulator. One interesting part of their model is the capability to simulate faults, which means that the model has consequence analysis capabilities. This is exemplified in their case study when faults occur in thrusters and generators, and they are able to simulate the response. Other use cases identified are optimization of emission-reduction, maintenance and fuel consumption, realistic load characteristics, concept

evaluation of new technologies, and more detailed thrust analysis. The concept evaluation is a particularly interesting use case for us, as it is listed as one of our current goals. The paper states *A simulation study should be the first step in evaluating the performance of new technical solutions, such as new energy storage components or new control strategies* (Bø et al., 2015). The reasoning behind this is that a simulation study is cost-effective and can give useful results without requiring too much resources.

The main contribution from this paper is the coupling between the electrical system and the vessel model together with the high level control. It goes deeper into the hierarchical breakdown of the ship systems and model them in more detail than this thesis will do. As we aim at doing long term simulations, it would perhaps only add to the inaccuracy if we were to model the short term dynamics as they do here, in particular with regards to the electrical system. In addition, simulating at this level of detail requires an almost complete vessel, in terms of specifications and machinery. This is also outside of our scope, which is to use simulation in the early stages of the design phase.

There are however many elements that we may learn from. An example of this is how they cope with the algebraic loop that occurs in the thrust allocation. The power that is available (i.e. produced and not yet allocated) from the generators is dependent on the power consumption, which in turn is constrained by the available power. They solve this by adding a lowpass filter on the power available signal, which is faster than the time scale of the power consumers. This may be valuable for us if we encounter algebraic loops.

Design, modelling and simulation of a hybrid fuel cell propulsion system for a domestic ferry

This paper by Bassam et al. (2016) uses numerical simulation to describe a hybrid fuel cell electric propulsion system on a ferry. First a model of the existing diesel propulsion system is built, and then validated by the actual operational data from the ferry. When this is done, the results are used to find the powering requirement from the new hybrid system, and the hybrid propulsion model is built, replacing the diesel propulsion system but keeping the rest of the model. In this paper they also justify the use of numerical simulation; *Numerical simulation is used in the design of the hybrid fuel cell propulsion system, because of the large cost of real testing and the lack of proper testing facilities* (Bassam et al., 2016).

A modular approach is chosen to build the ship simulator, similar to that of Bø et al. (2015). A representation of the model can be seen in figure 2.5. Each block performs a certain calculation

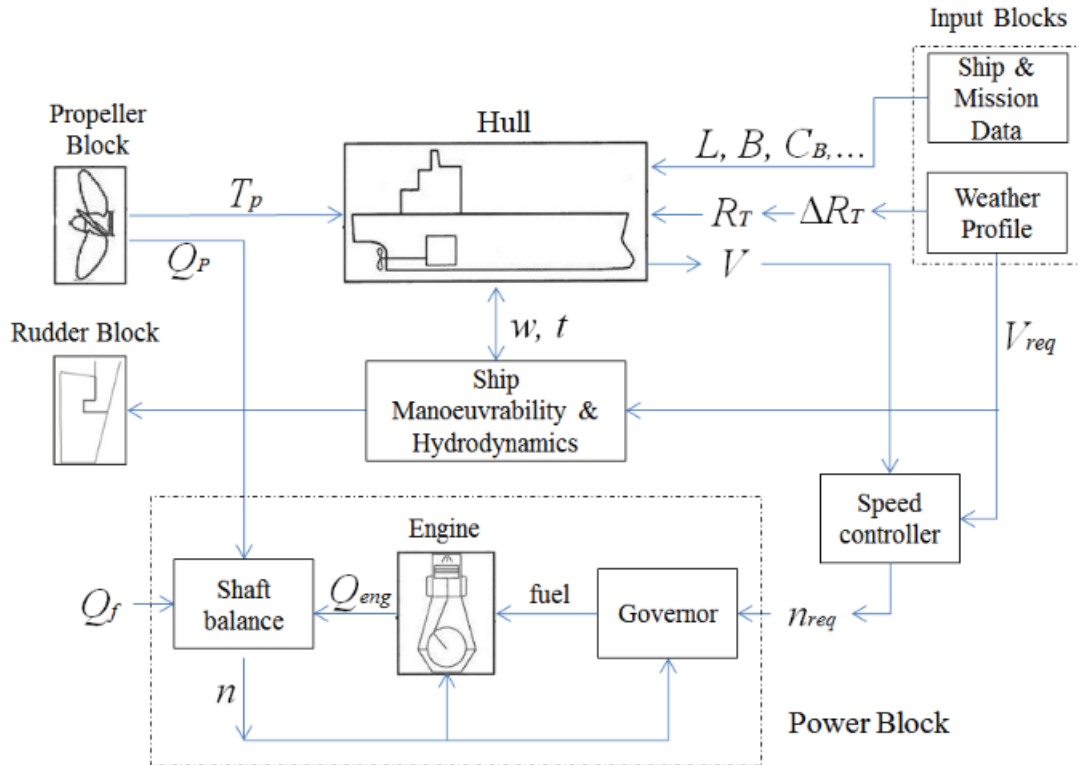


Figure 2.5: An overview of the total ship simulator by Bassam et al. (2016)

or represents a certain component of the system. At the top right corner of figure 2.5 we see how they provide data about the ship, the weather conditions and the mission data. In other words, this model needs to know the mission data in advance. This approach is good when you want to make changes to an existing system, because you can compare the different systems directly, using the same input data. However, if you want to test a different system altogether without having to provide so detailed input, we believe that this could be solved differently. One way of doing this is to replace the input blocks seen in figure 2.5 by a discrete event simulation model. This model should demand less input, for instance only an origin and a destination, and then virtually recreate the operating environment of the vessel (i.e. the external influences). We believe that merging this discrete event simulation model with a ship simulator as in figure 2.5 could result in a more self-driven total simulator, requiring less user input and hence be more applicable early in the design phase.

The ship simulator developed by Bassam et al. (2016) has three degrees of freedom (DOF), as opposed to most models which only have one DOF, i.e. ship speed. The reason for using three DOF is to give a more real representation of the performance of the ship. This could be relevant with regards to sea sickness, and in terms of operability when doing installation work or similar

activities. It could also give the model an edge with regards to capturing the effects of different loading conditions, as varying loading conditions could change the hydrodynamic responses of a vessel.

The data used in this model is gathered from the operation of a ferry over a period of two months. [Bassam et al. \(2016\)](#) explains how the raw data from the ferry can be pre-processed before being used in simulations. They choose to use windows with a size of three minutes. The size of the windows is based on the application, and is a trade-off between robust estimation and time-scale for change in the variables ([Bassam et al., 2016](#)). The data points in each window is then gathered, and they calculate the mean, variance and the derivative.

This approach with small time steps is dependent on having real operational data of sufficient quality, i.e. no large gaps or missing data. This is demanding both with regards to computational time, data-gathering and pre-processing, but it should give good results. We might use this method in a benchmarking or validation and verification scenario, if we get sufficient data from a real case. Otherwise our time steps will be decided largely by the time steps in our data sets. Most of the weather data, i.e. MetOcean, that we have available has a three hour time step. Consequently, we should model longer voyages for instance deep sea shipping, so that we are able to capture the fluctuations in operating conditions. Alternatively we have to model the weather data in a different way, so that we are able to capture the stochasticity and influence that the operating conditions has on the performance of the ship system.

Simulation of a Hybrid Marine Propulsion System in Waves

The paper by [Yum \(2016\)](#) uses simulation to test a hybrid propulsion system for a freight vessel. Hybrid propulsion systems are gaining popularity in the marine sector due to enhanced flexibility in operation and reduced risk, but it is still quite rare compared to conventional propulsion. Because this is still a novel concept, simulation is used for testing the feasibility and for initial design of the system.

Two challenges with system simulation models are identified, namely how to integrate the different models and how to run the overall simulation effectively. This can be solved in different ways, and one of the common approaches is to have all the models implemented in a single software tool. They find that this has some drawbacks, for instance with the incompatibility of different modules originating from different software, issues with intellectual property, and different time-scales between multi-discipline systems. Another way of solving this is to use a

distributed simulation technique.

A proposed way to solve some of these problems is to use co-simulation. Using co-simulation gives you a more flexible simulation environment, encapsulation of the component model for the protection of proprietary knowledge of original equipment manufacturers and increased possibility of distributed computing (Yum, 2016). This might be relevant for us if we get for instance engine models, and want to include them in the platform, without sharing the detailed information about the engines with other users.

In order to model the hybrid system to a sufficient quality so that the benefits of the hybrid system can be seen, it is crucial to understand the behaviour of the system under different operational conditions (Yum, 2016). This requires a complete system simulation with high-fidelity models. The biggest challenge identified by Yum (2016) is that the models of the subsystems have different complexity and a wide range of time scales dependent on their fidelity. This might make it harder to correctly model the interactions between the subsystems, and it will decrease the speed of computation if all is done by one solver.

In general, this paper has more detailed models of the vessels subsystems than we intend to have. The system simulator also deals with a wide range of time-scales, which is necessary in order to capture the dynamics of the various subsystems and modules. One of the reasons why this model has to be of this level of detail is the key performance indexes that the simulation is trying to measure. Amplitude of fluctuations of shaft speed and engine power, and voltage and frequency deviation are identified as KPIs, along with energy efficiency. These KPIs require insight and detailed modeling because they are dynamic and on small time scales (0.1 ms). This thesis will not go into such a detailed modeling, because we want to keep the modeling simpler, and have the components on the same general level of accuracy. However, as we want our simulation platform to be expanded in future work, we can learn from the framework and overall system description presented by Yum (2016).

Summary of Findings

Summarizing the ship simulation models described above, we see that all take a modular approach to building the model. It seems common practice that the ships are broken down into subsystems that are represented by blocks and sub models. This modular approach results in flexibility in terms of design, i.e. you can change components in the ship model quickly. It enables co-simulation, which can reduce simulation time, and "black box" thinking to protect

intellectual property across a simulation platform. Care should be taken with regards to the inputs and outputs of these modules, as there is a wide span of time-scales in the subsystems on a ship. This means that the interfaces between the modules have to be well defined and robust, and the completed models should be verified and validated against real system data.

We do not see a clear trend in terms of what scenarios are used for the simulations, or benchmarking. Some use historical data directly, and replicate the same run for different designs. This makes it easy to compare two similar designs directly, and is a viable approach if the mission of the ship is well defined. It is also useful for testing the effects of design changes to already operational ships. Others create different artificial scenarios, ranging from calm to rougher conditions. These scenarios can be based on statistics from the intended area of operation, or simply general sea states. This approach may give more "fair-grounds" for comparison, if for instance one design is specialized to a specific condition. It would perhaps be used when the mission of the ship over its operational life is not decided. On the other hand the results may be further from the real life operational performance of the final design.

There is a trade-off that has to be made between usability of the model early on in the design phase, and accuracy of the results; In the early phase, you don't know the specifics of your design, and therefore you cannot model to a high level of detail. As the design phase continues, the model can be improved and a higher level of accuracy achieved. The difficulty lies in balancing these two, as we want our model to be useful in both phases. A module based platform will allow for quick changing of the components describing the ship, so that the correct level of modeling detail can be attained. The trade-off between early stage applicability and accuracy in results is also influenced by the chosen KPIs to be investigated. Different KPIs demand different levels of modeling detail in order to capture the difference in performance. These KPIs are in turn decided by the type of operation that the vessel will do; For a tanker, fuel consumption might be the most important factor, while for a OCV the operability might be the most relevant.

3. Software

In this chapter we introduce the programs that are used in the thesis. Matlab and Simulink form the basis for the simulation model, while the ShipX workbench is introduced to perform hydrodynamic calculations. The theory behind ShipX is described and the formulae needed to use the output are shown.

The implementation of ShipX and description of the simulation framework is described in subsection 5.3.4.

3.1 MATLAB

Matlab is a matrix-based programming language that is used in a wide variety of science and engineering fields. Matlab also has a large library of prebuilt toolboxes and add-ons, from mapping and visualization to optimization and numerical mathematics.

In Matlab, scripts and functions that incorporate various add-ons and external programs can be executed. This allows the user to implement models and run them programmatically, meaning that you can run them without operating them separately. This gives flexibility in configuring models from the Matlab workspace, and results can be gathered and presented easily and rapidly.

3.1.1 Simulink

Simulink is a block diagram environment for multi-domain simulation and model design. Simulink has a large library of customizable building blocks that can be connected in a graphical editor. When a model is built, it can be included in parent-models in the same drag and drop manner as the built-in blocks. Simulink offers a plug-and-play feature for the building blocks, where it is easy to change one block with another, as long as the inputs match. It also has a good layout giving good insight into how the blocks interact.

Simulink models are incorporated with Matlab, and can read variables and data from the Matlab workspace. This allows the generation of data before running simulations, which in turn

allows the simulink model to be more generic and easily configurable. It also enables analysis of simulation results in Matlab, and easy storing and comparison of results.

3.1.2 SimEvents

Simulink is a time based system, i.e. the system state is defined at all times, but it has a discrete event component called SimEvents. SimEvents is in many ways very similar to Simulink; It has a library of built in blocks, and the same graphical editor where blocks are connected. The difference is that SimEvents is event driven, i.e. the system state is defined at the points in time when events occur. Simulink and SimEvents can also be combined into a so-called hybrid system, in which a discrete action may trigger a continuous calculation. For more information about Simulink, see [MathWorks \(2017b\)](#).

3.1.3 Batch Mode

Programming in Matlab gives the opportunity to run external programs through batch mode. In Matlab one can use shell escape to give commands to the operating system, making it possible to use external programs from a script or function in Matlab. One of the advantages of batch mode is that it decreases the amount of data that must be available before the start of a function, or in this case a simulation. The alternative would have been to process all the variables with all their respective ranges in advance, and then read from a matrix, this requires more of the user and computer as the number of variables increase.

3.2 ShipX

For the hydrodynamics module we have chosen to use ShipX for more exact calculations of resistance and vessel responses in different weather states. ShipX is a hydrodynamic workbench developed by MARINTEK (SINTEF Ocean), that includes several tools for hydrodynamic analysis. These tools use the same set of input data, which allows for integration with our simulation platform.

The purpose of this workbench is to make the analyses during a design process easier by gathering several necessary tools for calculation in the same program. ShipX has several plug-ins that can be applied depending on what the user requires.

The drawback for most of these plug-ins is that they are not compatible with batch mode. The only plug-in available for batch mode at the moment is Veres. As a vessel will sail and operate in a broad range of both significant wave heights and periods, it would be quite difficult to calculate added resistance for all combinations of the two, as well as wave heading. It has therefore been a priority to make Veres run from batch mode during the simulation.

For calculation of calm water resistance the plug-in Waveres is chosen, while the added resistance is calculated by the plug-in Vessel Responses.

3.2.1 ShipX Vessel Responses (Veres)

As the name suggests the Veres plug-in calculates vessel responses to waves. Veres can be used to calculate motion responses, global wave induced loads and operability at given sea states. The theory in the program is based on linear, potential, strip theory. Potential theory assumes a homogeneous, non-viscous and incompressible fluid, resulting in a negligence of viscous effects. The theory is valid for slender ships in moderate wave heights. Operating outside of the set limits has proven to give good results, but one should keep in mind that the accuracy decreases when doing so. (Fathi and Hoff, 2017)

Vessel data and condition information are needed to calculate added resistance. Further on, if one wants to look at operability for the vessel, sea-keeping criteria can be added, this is illustrated in figure 3.1.

To calculate added resistance in Veres there are two options. One can either use ordinary strip theory through Gerritsma and Beukelman's method, or direct pressure integration. Ordinary strip theory is applicable for monohulls at low or moderate forward speed. Direct pressure integration is available for all vessels and speeds.

Veres calculates added resistance operators as a function of ship speed and wave height and direction. From this operator it is possible to calculate the added resistance. The added resistance operator is non-dimensionalized as:

$$C_{RAW} = \frac{R_{AW}}{\rho g \zeta^2 B^2 / L_{pp}} \quad (3.1)$$

After running Veres one can use post-processors in ShipX to evaluate the results. There are four available post-processors; motions and operability, added resistance, global loads and general

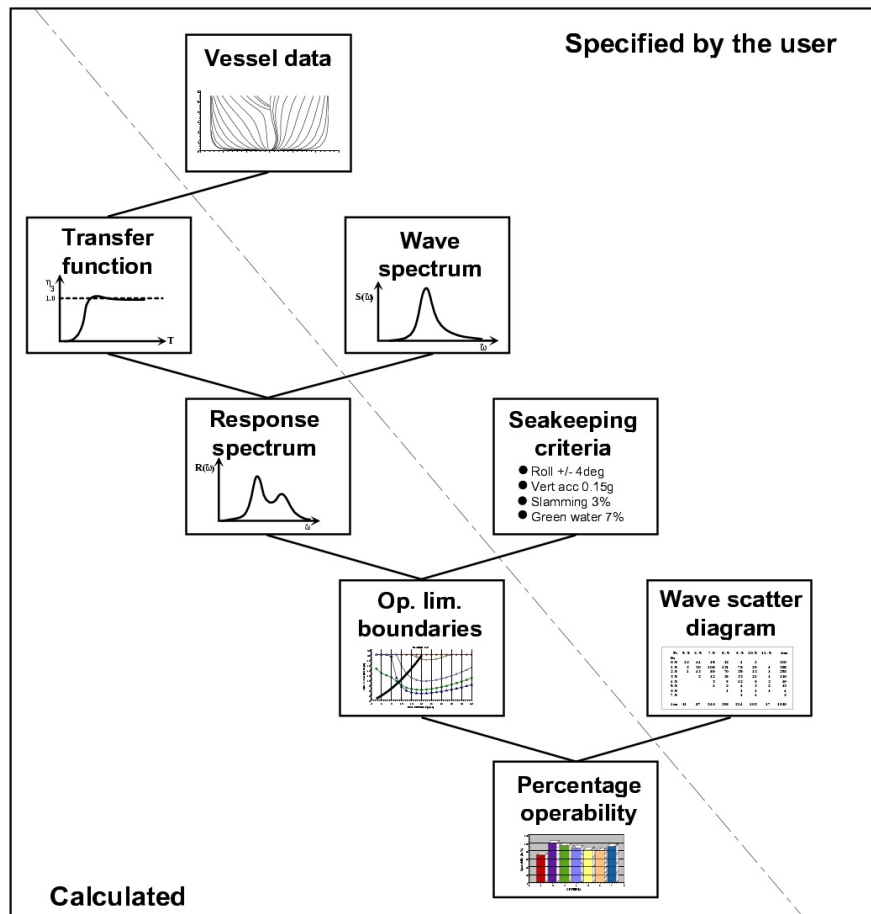


Figure 3.1: Input vs. calculated values from Veres (Fathi and Hoff, 2017)

response amplitude operators. The issue with the post-processors are that they are only available in ShipX, and not as a batch process. Meaning that the post-processor must be run manually for each individual run of Veres.

3.2.2 ShipX Wave Resistance (Waveres)

Waveres uses potential theory to calculate resistance, in this case; wave resistance. By combining linear and non-linear theory it provides reliable wave resistance estimates. Since Waveres (just like Veres) applies potential theory, it does not account for viscous forces. Viscous resistance can be calculated using empirical formulae. (Fathi and Steen, 2000) The following equations indicate how total calm water resistance is calculated.

$$C_W = \frac{\text{Wave Resistance}}{\frac{1}{2}\rho V^2 S} \quad (3.2)$$

Waveres also calculates a correction factor, F_{ds} , that accounts for the fact that the wetted surface at speed is larger than the nominal wetted surface due to sinkage.

$$F_{ds} = \frac{-\text{sinkage}L_{WL} * 2}{S} \quad (3.3)$$

Where the sinkage is calculated by Waveres. This correction is then used to calculate a modified viscous resistance coefficient in combination with the ITTC'57 friction formula.

$$C_V^{\text{Waveres}} = \frac{0.075}{(\log_{10}(R_N) - 2)^2} (1 + k)(1 + F_{ds}) \quad (3.4)$$

$$C_F^{\text{ITTC}} = \frac{0.075}{(\log_{10}(R_N) - 2)^2} \quad (3.5)$$

Finally, the residual resistance coefficient is found

$$C_R = C_W + C_V^{\text{Waveres}} - C_F^{\text{ITTC}}(1 + k) = C_W + C_F^{\text{ITTC}}(1 + k)F_{ds} \quad (3.6)$$

And the total resistance coefficient

$$C_T = C_R + C_F^{ITTC} \quad (3.7)$$

Resulting in the total calm water resistance

$$R_T = C_T \frac{1}{2} \rho V^2 S \quad (3.8)$$

4. Problem Description and Simulation Platform

This chapter will first present the problem description, and then describe the use of the simulation platform and for what purposes it can be applied. It will also give an overview of the different scenarios that can be simulated.

4.1 Problem Description

The objective for this thesis is to provide a designer with the tools to analyze a vessels performance during sailing, applying simulation. The goal is that the designer is able to evaluate a vessels performance subjected to close to real-life conditions based on location, external conditions and a vessel description.

This will be achieved by building a simulation platform that utilizes weather data to create the external forces acting on the ship. The platform includes a model of the ship and its key components, and the ship operation will be driven by discrete event simulation model (DES). The output of the simulation will be a time series representation of the ships performance, allowing the designer to analyze, compare, and draw conclusions about the design.

This will result in more precise prediction of the ships operational performance. It can also enable the designer to make better designs, by allowing him to assess the design performance in a more realistic operating context. It will also allow for rapid assessment of the design, and hence allow many iterations with various changes, making it more efficient in exploring the design space.

Table 4.1: Overview of the components in the simulation platform

Component	Description
Operations Module	A DES model of the operations. Simulates vessel operation, including sailing, and time spent in port. The operation is configured according to the route created in the master script.
MetOcean Module	A Matlab based model that gives the current weather conditions at the vessel position when the vessel is sailing, by using a Markov chain approach.
Ship Model	A Simulink model of the ship, consisting of a machinery and hydrodynamic module. It takes in the MetOcean conditions and the operational settings. Calculates the total resistance of the vessel, and powering need accordingly.

4.2 Introduction of the Platform

The constituent parts of the platform, and the flow of information between them is illustrated in figure 4.1, which will be explained in the following. The simulation platform is governed by a master script, written in Matlab. This script can be found in appendix B.1. The script consists of pre-processing of input data, execution of the simulation, and post-processing of the simulation output.

Pre-Processing

Starting at the top left corner of figure 4.1, we see the input that has to be supplied to the platform. The main particulars of the hull, data about the main engine and its fuel consumption is entered into an excel file called "Vessel Info". This file is read by the master script, and the information is made available for the simulation model as workspace parameters. The contents of this file can be seen in table B.1.

Weather data for the intended area of operation is supplied as historical hindcast data in excel files. These excel files are then pre-processed by the master script, prior to commencing the simulation. In the pre-processing, the hindcast data is used to create Markov chain transition matrices.

The pre-processing then generates a route based on a starting coordinate and a destination. A route can also be specified by the user, to deal with confined waters and navigational hazards.

Simulating

Once the master script has run through pre-processing, it initiates the simulation model. An overview of the modules and their functions is given in table 4.1. The initiation is illustrated by the transition to the top right part of figure 4.1. The DES model will assign the ship entity that moves through the model with attributes according to the information from the workspace.

When the entity is set to sail, the DES model calls for the MetOcean model to supply the current conditions. These conditions, along with the current attributes, are then sent into the ship model (represented by the Simulink model in figure 4.1). Here, hydrodynamic calculations are performed, along with calculation of the required power. The attainable speed in the current

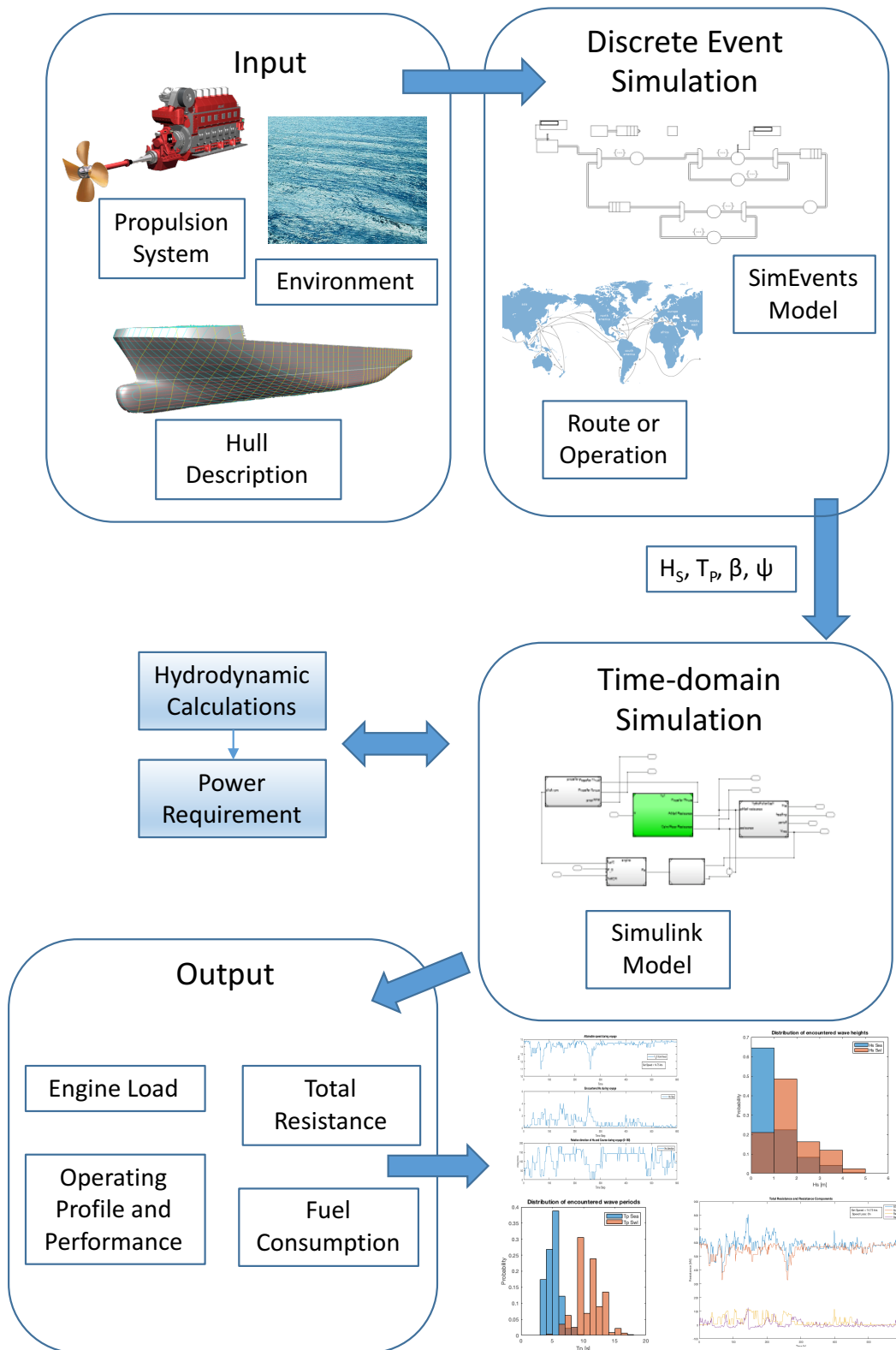


Figure 4.1: Overview of simulation platform

conditions are found, and the entity sails a distance equal to the attainable speed multiplied by the time step length. The position of the vessel is also updated. This sequence is repeated until the voyage is completed, at which point the simulation is terminated.

Post-Processing

The data from the simulations is stored at each point in time when the Time-domain simulation model is executed. This means that a value in the output is taken as the average value for that time step, and the performance of the ship is assumed constant over this time step. In the post-processing the output is assembled into a tabular format, and presented both visually and numerically. The output can then be stored, and a new run initiated.

Different Levels of Detail When Using the Platform

Based on how far along the design process has come, we have identified two main levels of detail for using the platform. The levels consist of factors that together describe the ship-ocean system. The purpose of these levels is to give the user some guidelines on how to best use the simulation platform, and how to interpret the results.

The first level is to be used at the very earliest design stages. Only main parameters are known and the systems aboard the ship are not fully defined. In this level, the platform should be configured to use the Hollenbach model for resistance calculations. Hind-cast weather data should be found, along a preliminary route. Regarding the output, emphasis should not be placed on the performance evaluation, i.e. break power need and fuel consumption. However, the MetOcean data, and the attainable speed following the speed loss, would still provide realistic values. Accordingly, sailing time and encountered weather can be used for further analysis.

The second level should be entered once a ShipX geometry file is available. This could for example be a comparison vessel model, using it to see how changes in dimension affects the results. Switching to the ShipX model for resistance calculations should provide more accurate performance evaluation, assuming that the geometry is correct. The systems aboard the ship, in particular the machinery, should be specified. Hind-cast weather data should now be gathered for points along the intended route. If this is done, the performance related output, i.e. fuel consumption and break power, should be more accurate. However, the results should be validated, either by running ShipX separately and comparing this to the results from the platform, or by comparing the results from the platform to real data.

Whichever level is used, we emphasize that the calculations are only as solid as its weakest link. Care should therefore be used when analyzing the results, and the results should be seen in context with each other. For instance, the time series plot of total resistance should follow the form of the plot of encountered wave heights.

4.2.1 Purpose

The purpose of the simulation platform is to support decision making by rapid prototyping and testing of ship designs. By using simulations, the platform is able to create a more realistic operating profile than today's spreadsheet calculations. The platform's key feature is that it can recreate years of operation very quickly, while also modeling the performance of the ship and its systems. This enables the user to optimize his design to the actual operating profile of the ship. We say that the operating profile is the actual one, because it is not created beforehand, but as a result of the virtual operation.

The platform runs the ship through a chosen scenario, and the ship's performance is recorded. This performance record gives the user a foundation from which he can further develop his design. The user can compare the performance of different designs, or he can make changes to the current design and test the new solution in the same scenario.

The ability to model weather (external influences), and handle the complex interactions between ship systems is another thing that separates the simulation platform from ordinary spreadsheet calculations. These calculations could obtain similar results, but it would take a lot more work from the user. The size of the program would grow to become over-complex and incomprehensible relatively fast due to the nature of the ocean-ship system, and how complex the interactions are between them. The platform yields a transparent overview of the system and the inner workings, combined with a fully working calculation that takes into account all the complex interactions between the sub-systems.

Another benefit for the simulation platform is the ability to expand the model and/or make the modules more complex. Improving a single module should not affect the other modules, at least not in a significant way, given that the output and input for each module are well chosen.

4.2.2 Rule Based Decision Making

Regarding speed and power there are three alternatives to how they can be represented, as presented by [Fathi et al. \(2013\)](#). Either, power is represented as a function of speed, and the input for the model is a constant speed that the vessel should keep. This would in a simulation sometimes lead to very large power requirements in rough weather periods where a captain in a normal situation would slow down, for then to speed up when the weather calms down. It is of course highly unrealistic that a vessel is able to keep the same speed at all times, and certain limits is therefore needed in a case where the vessel is required to maintain a given speed. An example of such a limit would be to implement an upper boundary for power requirement as a function of speed and weather conditions. The other alternative is to have speed as a function of power, so that the input for the model would be constant power. The model would then have to calculate what speed the vessel can maintain during all conditions so that the power target is maintained.

The third alternative, which is one of the options implemented in the model in this thesis, is to calculate attainable speed as a function of desired speed and weather conditions. One calculation method for attainable speed can be found in section 5.3.5.

4.2.3 Desired Output and Expected Interpretation

There are several parameters desired as output from the model, some of them have been implemented in the model, while others are parameters that should be implemented in future work. A common feature for all outputs from this model is that they have a level of uncertainty, meaning that a ship model exposed to the same circumstances in the simulation model as a ship in real life will most likely not produce the same results.

Desired output parameters are to some degree based on what type of operation the vessel is set to do. During a DP operation the operability factor is for example more relevant than the expected fuel consumption. Operability is mostly affected by weather conditions and is a factor implying how much of the expected work time the vessel actually can work. As far as fuel consumption is concerned it will always be a relevant factor, independent of operation. Now more than ever, fuel consumption and chosen type of fuel is of increasing interest among ship owners. This is mostly due to the increasing concern for the environment and the impact shipping has. A green profile is required from IMO and other maritime regulators, as well as it can help boost

the public opinion of a ship owner.

As long as the user is aware of the shortcomings related to the accuracy level, there is definitely a good use for the output of the model. While the accuracy level is imperfect, it will give indications towards how well vessels perform relative to each other. This gives us the opportunity to compare vessels and see how they perform on the same tasks.

4.3 Scenarios for Simulation

4.3.1 Sailing

The first scenario that we are able to simulate is sailing. For most ships, sailing is the most important aspect of operation and it is critical to be able to predict factors such as powering need and attainable speed.

If the trade is known, for instance for a cargo vessel, then the coordinates of the ports that the vessel will visit is applied. The route can then either be generated automatically, or specified manually. If the route is in confined waters, care should be taken to ensure that the vessel follows the correct route, i.e. avoid grounding or collision.

The simulation aims to replicate the actual operation of the ship. First the route is determined in pre-processing. Based on the route, MetOcean data can be gathered to give a realistic environment. In the model, the virtual ship is built up of modules. The vessel is then set to sail at a certain speed, and the calm water resistance is found. Also the added resistance due to wind and waves can be found based on MetOcean data for the current position of the vessel.

4.3.2 Round Trip

A round trip scenario can also be simulated by the platform. In this scenario, the vessel operates between two ports. The discrete-event simulation part of the platform can model port operations, and different loading and unloading times.

4.3.3 Heading and Route

In order to give the ship its heading and route, and facilitate the gathering of MetOcean data, we use the mapping toolbox provided by Matlab. This toolbox contains maps and geographical data, and allows us to plot the voyage as well ([MathWorks, 2017a](#)). The purpose of the routing and scheduling section, is to find the route that the vessel should follow, based on the vessels current location and its destination. With the route mapped, the route can be divided in legs where MetOcean data is extracted for each leg.

The shortest path between two points on the earth is along a great circle. A great circle of a sphere is the intersection of the sphere and a plane that passes through the center point of the sphere. Accordingly, the shortest route between two points in the ocean is along a great circle line. However, it's impractical to follow a great circle course, because you would have to shift heading continuously. The other option for navigators is to follow a rhumb line. A rhumb line is a line of constant heading, resulting in simpler navigation as you only have to set the course once. The downside of following a rhumb line is that the voyage will be longer, because of the curvature of the earth.

One way of solving this is to approximate the great circle by rhumb lines, and in fact you don't need many segments to get close to the shortest route ([MathWorks, 2017a](#)). The number of line segments used to approximate the great circle is decided by the number of waypoints used. We illustrate this in figure 4.2, where you see the great circle in red with a distance of 4750 nautical miles, and the rhumb line in green with a distance of 5015 nautical miles. In other words, the great circle route is about 5% shorter than the single rhumb line. The blue line in figure 4.2 is made up of three rhumb lines, i.e. there are two waypoints. It has a total length of 4790 nautical miles, which is less than 1% longer than the great circle, which is the optimal route. When you consider that you only have to set the course three times, and the reduced costs involved, it becomes apparent that this approximation works well.

The approach described above is implemented through the function described in appendix B.2.1. The function does not recognize land from sea, so it should be used with care. Waypoints can also be written manually for more complex navigation, for instance when approaching a port.

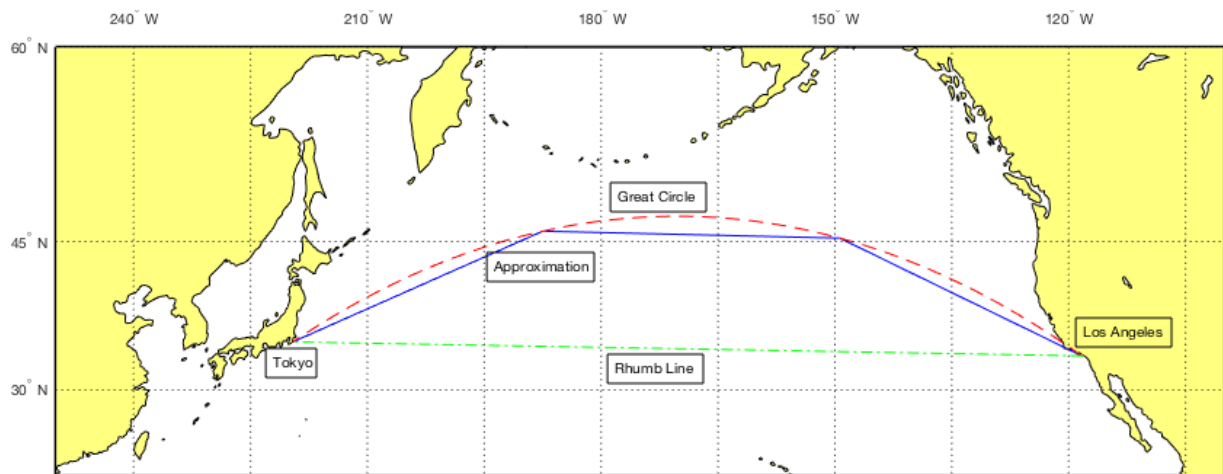


Figure 4.2: Illustration of great circle and rhumb lines, between Tokyo and L.A.

5. Simulation Platform and Modules

This chapter describes the modules that come together to form the simulation platform. The operation module simulates the operation of the vessel, i.e. sailing and port calls. The MetOcean module provides the environmental conditions, i.e. waves and wind, based on input from the operation module. Finally, the hydrodynamics and machinery modules calculate the vessel performance in the given conditions.

The theory and methods used in each module will be explained where they are used. At the end of the chapter, the modules are assembled to show the complete simulation platform.

5.1 Operation Module - Simulation of Vessel Operation

The vessel operation module is built in the event based SimEvents environment (see section 3.1.1). This module runs the vessel through its route, and initiates the other modules to perform calculations.

In SimEvents, the vessel is represented by an entity. The entity contains information about the ship, in the entity attributes. These attributes can be read and modified by the other parts of the platform through entity servers. When the entity arrives at a server, the server can read the attributes and call on functions that perform calculations. Table 5.1 describes the attributes used in the operation module.

Two different models have been developed in this thesis, one sailing from a to b, and one sailing in a round trip between these points. The first model lets the user evaluate sailing performance only, which reduces the computational time and highlights vessel performance. The round trip model includes ports at points a and b, and shows the user the different performance when sailing the return leg, because of differences in encountered MetOcean conditions. This model also allows long run simulations, for instance over a year, including the full spectrum of MetOcean conditions.

Table 5.1: Description of the attributes of the ship entity

Attribute Name	Unit	Description
HsSea_S	[-]	State of the Markov chain for wind based Hs
HsdSea_S	[-]	State of the Markov chain for wind based wave direction
TpSea_S	[-]	State of the Markov chain for wind based wave period
HsSwl_S	[-]	State of the Markov chain for swell based Hs
TpSwl_S	[-]	State of the Markov chain for swell based wave direction
U10_S	[-]	State of the Markov chain for wind speed 10 m above sea level
U10d_S	[-]	State of the Markov chain for wind direction
Speed	[kts]	The speed that the vessel is set to operate at
Outputport	[-]	Directs the vessel at output switches
Leg	[-]	Keeps track of what leg of the voyage the vessel is in
Legstep	[-]	Keeps track of steps within a leg
Legdist	[nm]	The distance of the current leg
Course	[deg]	Course of the vessel
Lat	[deg]	Latitude of the vessel
Lon	[deg]	Longitude of the vessel
Hs4k	[m]	The Hs to be used in speed loss calculation
Hsd4k	[deg]	The wave direction to be used in speed loss calculation
Att_speed	[kts]	The attainable speed at current conditions
Dist_sailed_tot	[nm]	The total distance sailed during voyage
Dist_sailed_leg	[nm]	The distance sailed on the current leg
NumRTs	[-]	The number of round trips currently completed
Voyage_step	[-]	The current part of the overall voyage

5.1.1 SimEvents Sailing Model

The sailing model can be seen in figure 5.1. At the top left part of this figure, the entity is generated. Only a single entity is generated, at time zero. Once the entity is generated, it moves into an entity server that sets the starting values of different attributes. These attributes are read from the Matlab workspace via a simulink function.

Once the starting attributes are set, the entity moves into the sailing loop via an entity input switch. In this loop, another simulink server (Run simulink) initiates a call to the MetOcean module (section 5.2). The current state of each Markov chain, along with course, speed, and current leg, is included in this function call and sent to the MetOcean module. The function call returns new states for the Markov chains, along with the actual wave height and direction. The states are stored to be used in the next function call. The wave height and direction is then used to calculate the attainable speed via a simulink function. The attainable speed is then used to calculate how far the vessel sails in one time step, and the new coordinates of the vessel is found. The entity then leaves the Run simulink server.

In the next server in the sailing loop, the total distance sailed is checked. If the destination is reached, the entity is sent to a terminator and the simulation is stopped. If the destination is

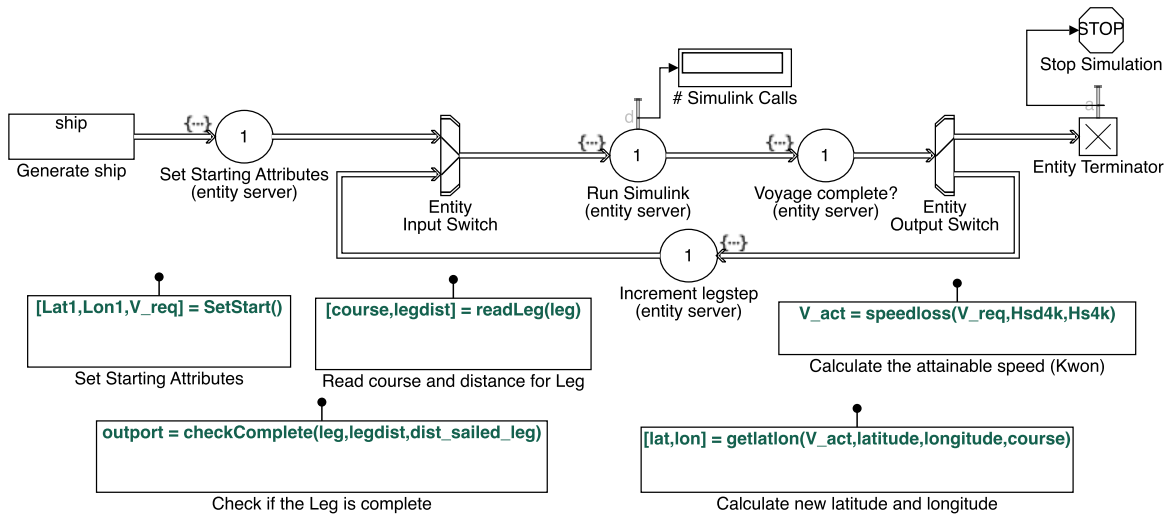


Figure 5.1: SimEvents sailing model

not reached, the entity is sent back to the input switch, and the legstep is increased. When the legstep is increased, the model also checks what leg of the voyage the vessel is at. If the distance of the current leg is reached, the leg counter is increased. This sequence is repeated until the entire voyage is complete and the simulation is stopped.

5.1.2 SimEvents Round Trip Model

The round trip model, seen in figure 5.2, includes two sailing loops. The first part of the round trip model is similar to the single sailing model. The entity is generated in the same way, with the same attributes. However, this model also includes port operations. The port operations is modeled by entity servers, see Port A and Port B in figure 5.2. The port servers initiates function calls to the MetOcean module in a similar manner to the sailing servers (Run Simulink). The difference is that the speed is set to zero, so that no power is produced when the vessel is in port. The service time, i.e. time to load and unload, can be fixed, or set to be a function of the vessels cargo capacity. It can also be drawn from a probability distribution, to imitate delays or general variance in the time spent in port.

After the first port call, the vessel enters a sailing loop. This loop works as described in section 5.1.1. Once the vessel has sailed the entire distance from port A to port B, it begins its port call at port B. After the completion of this port call, it enters a second sailing loop. In this loop, the route is flipped, and the vessel sails back to port A. Upon reaching port A, the entity enters a

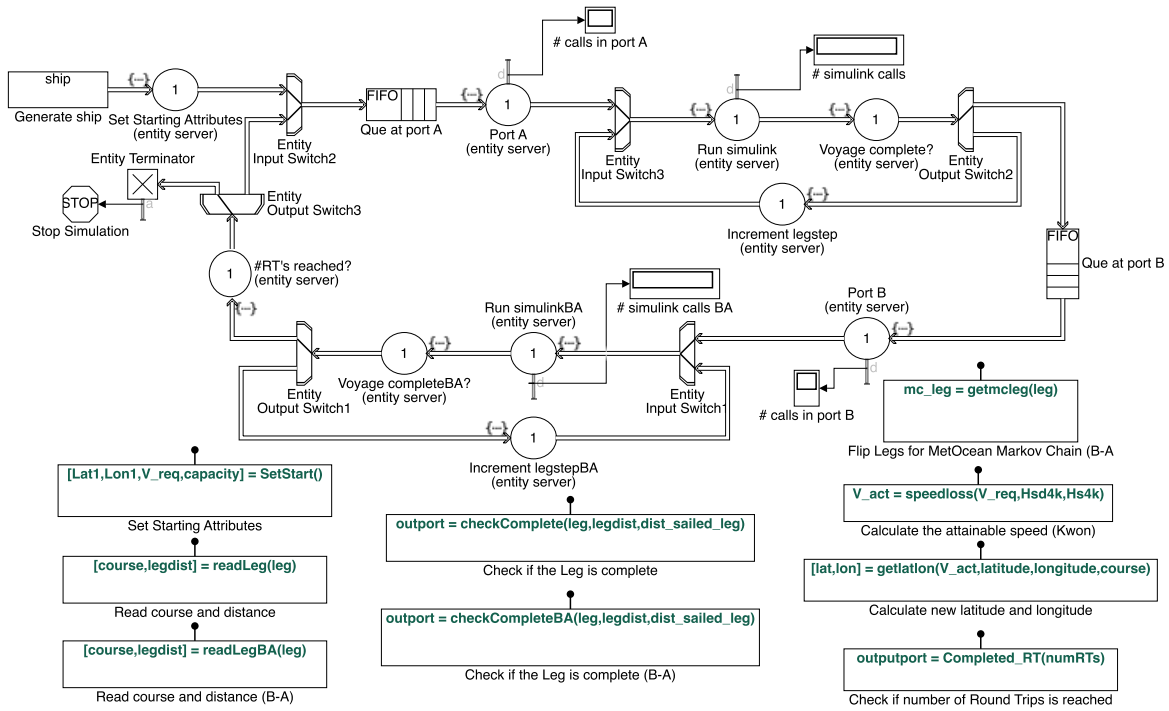


Figure 5.2: SimEvents round trip model

server that counts the number of round trips completed. If the desired number of round trips is reached, the entity is sent to a terminator and the simulation is stopped.

5.2 MetOcean Module - Simulating Operating Conditions

The MetOcean module, shown in figure 5.3, is an important part of the platform. It is this module that incorporates realistic environmental conditions, and in many ways this is what differentiates the simulation approach from more straight forward performance evaluation methods.

The MetOcean module is placed within a Simulink function, which is called when the vessel is sailing or at port, see section 5.1.1. The MetOcean module reads where in the voyage the vessel is and which course it has. Together with the vessel data and the last known conditions, i.e. previous states, the module is able to draw new states resulting in new conditions. Lastly, the module calculates the relative direction between the waves and wind, and the course of the vessel. The conditions that are included in the MetOcean module are shown in table 5.2.

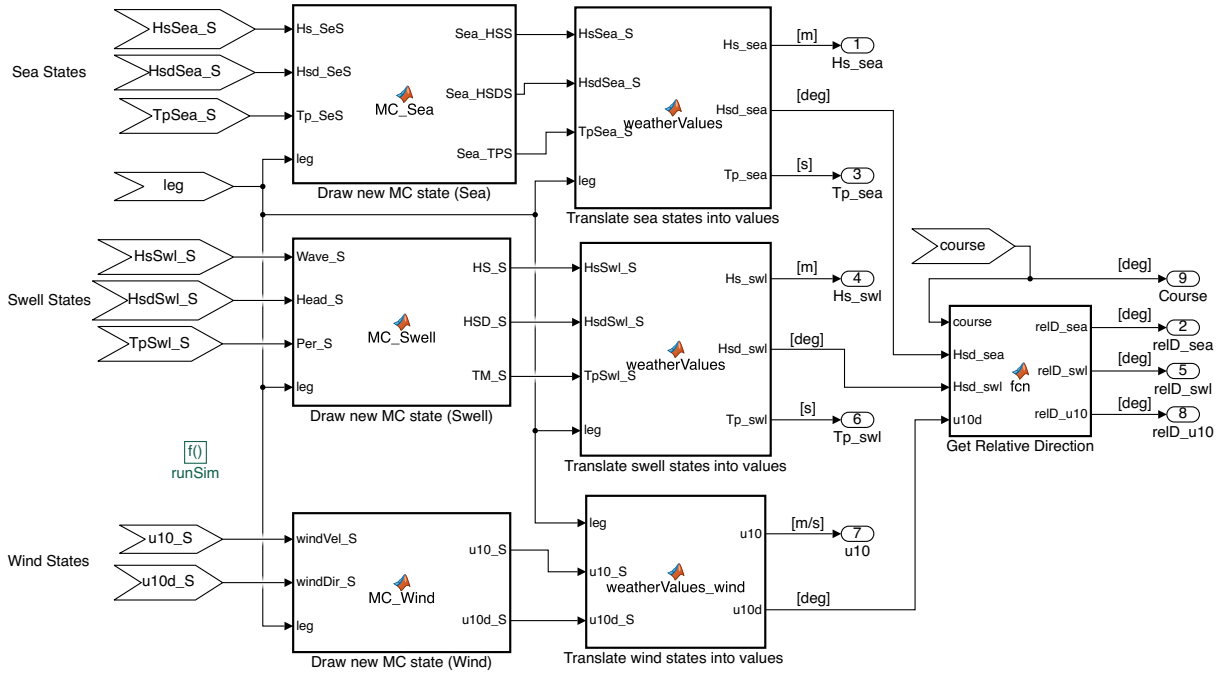


Figure 5.3: MetOcean module

Table 5.2: Output from the MetOcean module

Name	Unit	Description
Hs Sea	[m]	Significant wave height of wind based waves
RelD Sea	[deg]	Wind based wave direction relative to the course
Tp Sea	[s]	Wind based wave period
Hs Swell	[m]	Significant wave height of swell based waves
RelD Swell	[deg]	Swell based wave direction relative to the course
Tp Swell	[s]	Swell based wave period
U10	[m/s]	Wind speed 10 m above sea level
RelD U10	[deg]	Wind Direction relative to the course

Table 5.3: Markov chain transition matrix for significant wave height in the Pacific

State	1	2	3	4	5	6	7	8	9
1	0.85	0.14	0.01	0.00	0.00	0.00	0.00	0.00	0.00
2	0.15	0.70	0.14	0.01	0.00	0.00	0.00	0.00	0.00
3	0.01	0.23	0.62	0.13	0.01	0.00	0.00	0.00	0.00
4	0.00	0.05	0.25	0.54	0.15	0.01	0.00	0.00	0.00
5	0.00	0.00	0.05	0.30	0.50	0.14	0.00	0.00	0.00
6	0.01	0.01	0.02	0.08	0.26	0.44	0.18	0.01	0.01
7	0.00	0.00	0.01	0.03	0.14	0.23	0.53	0.07	0.00
8	0.00	0.00	0.00	0.00	0.00	0.12	0.29	0.47	0.12
9	0.00	0.00	0.00	0.00	0.00	0.00	0.25	0.13	0.63

5.2.1 Stochastic Approach to Modeling Weather Conditions

With a historical database containing weather data for given locations along the route it is possible to make a statistical distribution. From such a distribution it is possible to sample random weather conditions at given time intervals. By applying Markov chains, stochasticity is introduced to the model. The MetOcean module requires historical weather data for the relevant area, preferably divided into legs of the journey. Then the module will generate an individual Markov chain with respect to each leg and the relevant weather parameters.

The Markov chain approach allows a memory in the model where the next condition will depend on the previous one, using the transition matrices created from the historical data. This ensures a more realistic development in weather along the route, without the memory property the weather could change from calm sea during one cycle in the model and to extremely rough in the next, which is improbable in the real world.

An example of a transition matrix with nine states is presented in table 5.3, with the values that each state represent in the subsequent table 5.4. During generation of the Markov chains it is imperative that absorbing states¹ are avoided, since there is no such thing as an absorbing state when it comes to weather conditions. The values are based on the maximum and minimum values in the data set, where the range is divided into equally spaced steps, with a number of steps corresponding to the number of states plus one. Then each value is calculated as the mean for two sequential states, i.e. the value for state 1 is equal to the minimum value plus half the step value.

The values chosen for the states are the mean values between two sequential states in the valid range, i.e. min to max. The code for MC generation is found in appendix B.3.1.

¹An absorbing state in a Markov chain is a state that cannot be left.

Table 5.4: Values corresponding to the respective states found in table 5.3

State	1	2	3	4	5	6	7	8	9
Significant wave height [m]	0.46	1.36	2.25	3.15	4.04	4.94	5.84	6.74	7.63

5.2.2 Random Number Generation

One of the key elements in simulation is to have a routine to generate random numbers or values, for variables. Commonly, random numbers are generated in two steps (Jain, 1990). First, a sequence of random numbers distributed uniformly between zero and one is obtained. Then the sequence is transformed to produce random values satisfying the desired distribution. The first step is called *random-number* generation, and the second *random-variate* generation.

By nature, none of the random number generators are completely random; given the same seed they will provide the same results for every iteration. It is therefore important to keep track of seeds used, assuring the user that the model is stochastic and probabilistic. A *seed* is the number used to initialize the random number generator, and this has to be supplied to the generator by the user.

In the platform random numbers are used in the Markov chain approach to weather modeling. Here, the random number effectively decides which state the chain transitions into. Matlab has a variety of random number generators available, but in Simulink these are severely limited. We deal with this by creating a large quantity of random numbers in the Matlab workspace before running the Simulink model. The model can then use these numbers when deciding which state to transition into in the Markov chain. This allows us to control the sequence of random numbers generated, and store it, i.e the seed, in order to make an identical run, or change it to create different runs. The code used to determine the next state of the Markov chain is shown in appendix B.3.2.

5.2.3 Wave Spectrum

A quick glance out to sea will show you that the ocean does not follow a sinusoidal path. In practice engineers often use wave spectres to describe the seas. A wave spectrum describes the distribution of energy in a sea state, and contains statistical information about the surface of the sea (Pettersen, 2007). The parameters needed to create a wave spectrum can be found from advanced wave measurements, but these are often not available for the designer. A solution is

to use standardized wave spectres, dependent on given wave parameters to approximate the desired conditions (Pettersen, 2007).

When a spectrum is generated there are several assumptions made. First of all, it is assumed that the operation is in open waters so that the wind is not affected by land formations. It is also assumed that the energy in waves is constant at high frequencies, yielding that the energy as a function of frequency goes to zero when the frequency approaches infinity. There are several known wave spectres, two of the more famous are the JONSWAP spectrum and the Pierson-Moskowitz spectrum (PM). JONSWAP is an acronym for Joint North Sea Wave Project and is developed for the south-east part of the North Sea. The spectrum is based on the PM spectrum, but with an extra peak enhancement factor included (Myrhaug, 2007). A third option for the spectres is Torsethaugen which combines wind-sea and swell to calculate a double peaked spectrum.

5.2.4 Processing Weather Data

Weather data was downloaded from Waveclimate (BMT ARGOSS, 2017), where the parameters available can be seen in table 5.5. The first approach to weather data was to use H_s , H_{sd} and T_p (without suffix), which describes the combination of wind-sea and swell. The combined sea values for significant wave height are calculated as follows:

$$Combined\ Sea = \sqrt{(WindSea)^2 + Swell^2} \quad (5.1)$$

In figure 5.4, Vessel 1 (described in 6.5) has been used to find the mean added resistance as a function of wave period with a significant wave height of one meter and head seas. From figure 5.4 it can be seen that the swell values with the greatest probability for peak wave period (T_p) has close to identical peak period as for the added resistance. While both wind-sea and combined sea are shifted to the left. Using values for combined seas in this case would yield smaller resistance should the wave have a period of 14 seconds, for example. This would be a big source for error during added resistance calculations. This is why we have chosen to calculate the added resistance for swell and wind-sea separately, and then add them together to obtain the total added resistance.

Table 5.5: Parameters available from Waveclimate

Code	Explanation
u10	One hour wind speed at 10m asl
u10d	Wind direction at 10m asl (nautical)
Hs	Significant wave height
Hsd	Mean wave direction
Tz	Zero-crossing wave period
Tm	Mean wave period
Tp	Peak wave period
sea	Suffix for wind-sea
swl	Suffix for swell

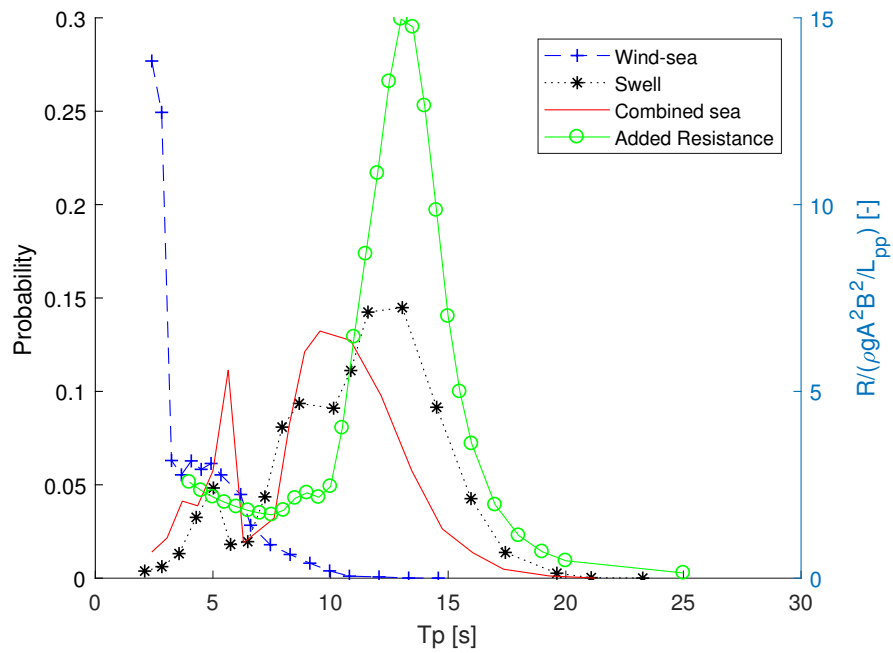


Figure 5.4: Empirical prob. dist. for peak wave period during head seas, mid-pacific (BMT ARGOS, 2017)

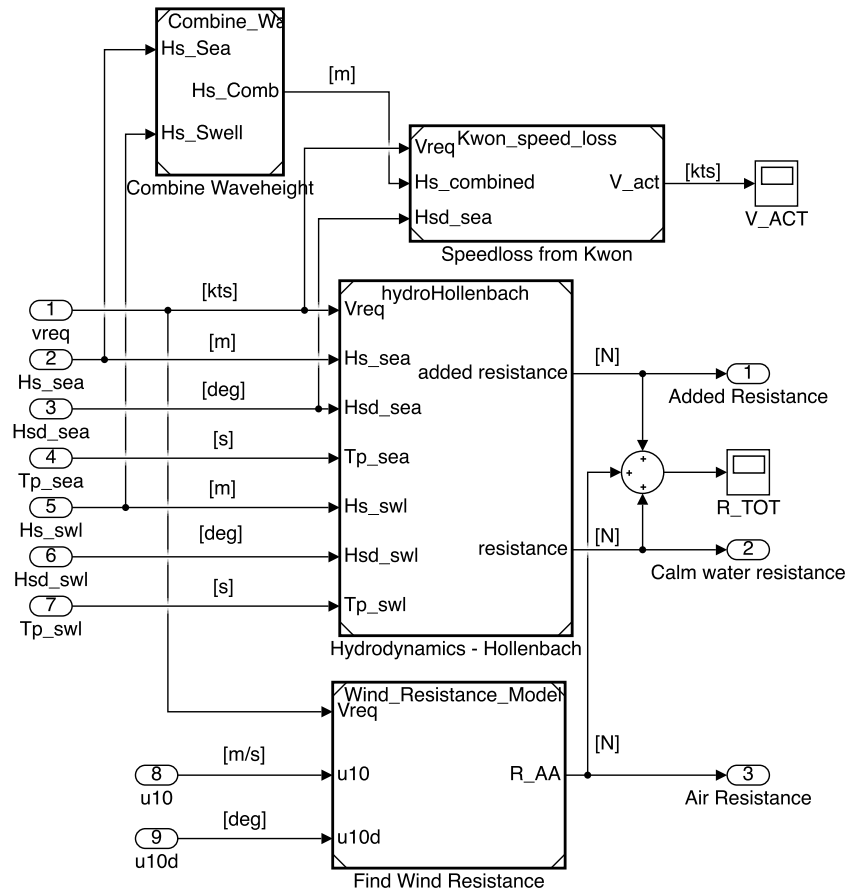


Figure 5.5: Hydrodynamics module

5.3 Hydrodynamics Module - Ship Resistance

The main task for the hydrodynamics module is to calculate the total resistance for the vessel at any given time, depending on external forces such as weather conditions, and chosen speed. The hydrodynamics module is initiated when input is provided from the MetOcean module, i.e. at every time step when the vessel is either sailing or in port.

The hydrodynamics module can be seen in figure 5.5. The input to this module is the same as the output from the MetOcean module (table 5.2), in addition to the set speed. The module then calculates the total ship resistance, and the attainable speed under the current conditions. In the following sections, the underlying theory for these calculations will be shown.

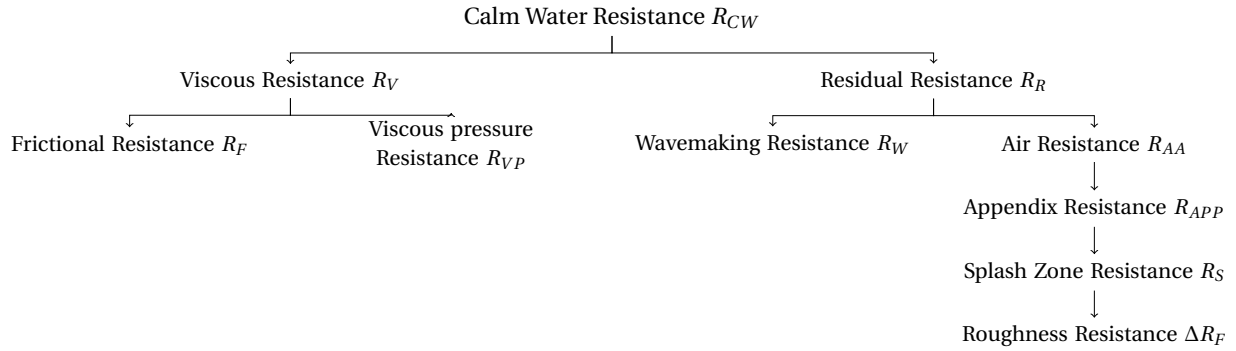


Figure 5.6: Subdivision of total resistance into components, for a vessel sailing in calm seas. Based on [Amdahl et al. \(2005\)](#)

5.3.1 Ship Resistance

The total resistance of a ship can be divided into three components

$$R_{TS} = R_{CW} + R_{AW} + R_{AA} \quad (5.2)$$

where R_{CW} is the calm water resistance, R_{AW} is the added resistance in waves, and R_{AA} is the air resistance.

Calm Water Resistance

The calm water resistance has two main components, residual and viscous resistance. A hierarchical breakdown of the calm water resistance can be seen in figure 5.6. Figure 5.7 shows a typical calm water resistance curve for a displacement ship. The figure illustrates how the composition of the total resistance changes with increasing speed. We see that wavemaking resistance is the largest part of the residual resistance, and that its share of the total increases with increasing speed. Further explanation of R_{CW} can be found in [Pettersen \(2007\)](#) or [Steen and Minsaas \(2013\)](#).

Added Resistance in Waves

The added resistance in waves represent the wave induced loads on the ship. The added resistance is mostly due to absorption of wave energy and damping forces due to additional acceleration. When the wavelength of the incident waves are smaller than approximately half the

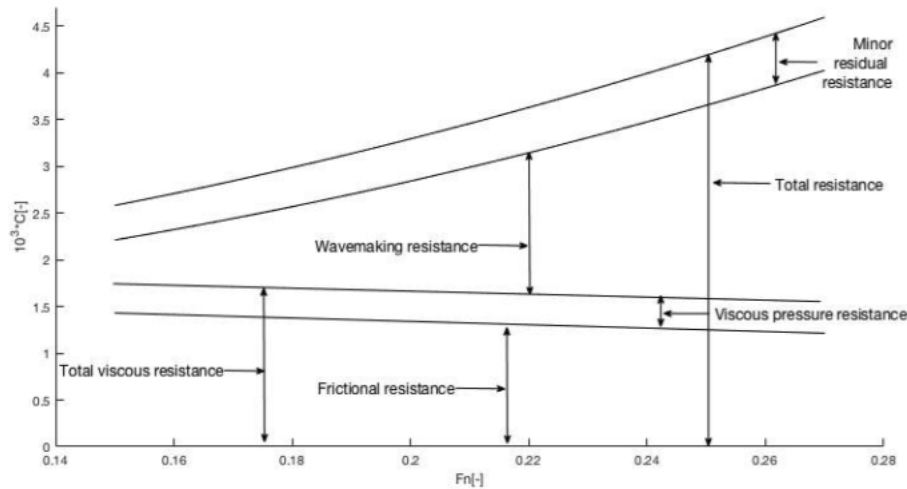


Figure 5.7: A typical resistance curve showing the composition of the total resistance coefficient for different Froude numbers for calm seas

ship length, the wave induced ship motions are small and the added resistance is mostly due to wave diffraction caused by the ship. When the ship motions become larger, they strongly influence the added resistance in waves (Prpic-Orsic et al., 2016). Further explanation of the added resistance in waves can be found in for instance in Faltinsen (1993).

It can be hard to estimate the added resistance, and typically a sea margin (as described in section 2.1) is used to account for the added resistance due to wind, waves, currents, fouling of hull and propeller and other factors. Prpić-Oršić and Faltinsen (2012) found that a 175 meter long container ship operating in the North Atlantic, showed an average yearly increase of CO₂ emissions of 15% compared to calm sea emission. This serves as an example of the effects of added resistance when operating in real conditions.

Air Resistance

The aerodynamic resistance, or air resistance, depends on the surface area of the ship over the waterline, and the relative wind velocity. The air resistance is calculated as follows

$$R_{AA} = C_{air} \frac{\rho_{air}}{2} (V_{rel} |V_{rel}| - U^2) A_p \quad (5.3)$$

where C_{air} is the air resistance coefficient for the superstructure, with typical values ranging

from 0.5 to 0.8. U is the ship velocity in meters per second. A_p is the transverse projected area of the superstructure and V_{rel} is the relative wind velocity (Steen and Minsaas, 2013).

5.3.2 Resistance Prediction

Accurate prediction of ship resistance is important in order to obtain the required propulsion power, which in turn is important for the selection of propeller configuration. The total resistance of a ship moving steadily through water is the component of the total force acting in the direction opposite to the movement of the ship. As described in the previous section, there are many contributions to the total resistance. Some of the contributions are difficult to obtain due to the relatively complex three dimensional flow pattern around the ship (Steen and Minsaas, 2013).

In general, one can divide resistance prediction methods into three categories, following Steen and Minsaas (2013); Purely empirical methods, numerical methods and methods based on Froude's hypothesis. Purely empirical methods are based on practical experience, and often uses ships of similar design to predict the resistance. This might be sufficient for ship owners and designers in an early phase, but it does not add anything of interest to this thesis and is left out.

Resistance prediction using Froude's hypothesis can be divided into three sub-categories; model testing, empirical, and numerical. Model testing should be done with the finalized design to get an accurate prediction of the resistance and the flow around the hull and propeller. This is a time consuming and expensive procedure and not applicable for our simulation platform. The two other methods are implemented into the platform. Resistance prediction using Froude's hypothesis will be explained in section 5.3.3, and numerical methods are investigated in section 5.3.4.

The two resistance prediction methods are developed in two separate models that can be used to perform the hydrodynamic calculations. The first approach is based on Hollenbach's resistance calculations, while the second approach is based on using ShipX to calculate the resistance. The two different approaches are included in order to accommodate use of the simulation platform both at the very beginning of the design process, and once a hull geometry is established. The decision on which module to use should be based on the desired overall level of detail, described in section 4.2

5.3.3 Hollenbach Model

Hollenbach's method for resistance prediction is based on a regression analysis of more than 400 ship models. It is also fairly new compared to other similar methods, and is quite easily implemented into a computer program. Hollenbach's method is also more accurate than other methods such as Holtrop and Guldhammer-Harvald (Steen and Minsaas, 2013). It should be noted that this method does not include bulbs, and that the method originally did not include form factors. Hollenbach's method has a standard deviation in the estimated resistance ranging from 9,4% to 13,3% depending on the number of screws and draft condition (Steen and Minsaas, 2013). In other words the deviations are too large to be used elsewhere than early in the design phase, and this should be kept in mind when assessing the results.

The Hollenbach calculations does not include added resistance in waves, or air resistance. Therefore we apply a formula by Kreitner, that estimates the resistance increase in waves, as suggested by ITTC (2005). This formula is valid only for wave heights up to two meters, and has been shown to have large variations in terms of accuracy (ITTC, 2005).

$$\Delta R_T = \frac{0.64\zeta_w^2 B^2 C_B \rho_w g}{L} \quad (5.4)$$

If the wave height is larger than two meters, the added resistance due to waves is taken as 20% of the calm water resistance. Finally, the Hollenbach model uses equation 5.3 to calculate the air resistance. The code that runs the Hollenbach model, including the formulae described above, can be seen in appendix B.4. The code that runs the original Hollenbach method is provided in part by (Steen and Minsaas, 2013).

5.3.4 ShipX Model - Veres and Waveres

Numerical resistance calculations are done by implementing the ShipX plug-ins Waveres and Veres into the platform. Both of these plug-ins require a ShipX geometry file. Waveres is used to calculate the wave resistance, and Veres is used to calculate the added resistance in waves.

Waveres

Since Waveres is not compatible with batch mode, it is run in pre-processing. Waveres calculates resistance due to wavemaking, which is only dependent on ship speed. The way it is done is that the user does a wave resistance run in ShipX, and exports the result file with resistance coefficients. The master script uses these coefficients to make a table with resistance depending on vessel speed. To calculate wavemaking resistance, interpolation between the results are used for when the speed is between the calculated speed values. Waveres is run for every whole or half knot, which makes the interpolation accurate enough for this simulation, especially when desired speed is used as input, as it is usually given as half or whole. The speed range and increment can be defined by the user. Later the empirical formulae described in 3.2.2 are applied in order to find the total calm water resistance.

Veres - Added Resistance

Added resistance is dependent on both weather conditions and ship speed, therefore this is preferable to calculate during the simulation. Calculating in advance would require calculations for a wide range of values for both wave periods, wave heights and wave direction, and then during simulation one would interpolate between the three. Thus, added resistance is calculated for every new condition in every iteration of the simulation through batch mode, explained in chapter 3.

The alternative to ShipX's post-processor for added resistance, where short term statistics is applied, is to write a code in Matlab that can produce a response spectrum for mean added resistance as a function of wave period. This is done using a wave spectrum combined with the added resistance operator from Veres. The mean added resistance, $\overline{R_{AW}}$, in an irregular seaway is calculated (Fathi et al., 2012)

$$\overline{R_{AW}} = 2 \int_0^{\infty} S(\omega) \left(\frac{\overline{F_1}(\omega; \beta)}{\zeta^2} \right) d\omega \quad (5.5)$$

where $S(\omega)$ is the wave spectrum² and $(\frac{\overline{F_1}(\omega; \beta)}{\zeta^2})$ is the added resistance operator read from the Veres results. The added resistance operator is dependent on wave frequency, ω , and wave direction, β , relative to the ships direction (Fathi et al., 2012). Waves from 0 degrees are head seas,

²Subsection 5.2.3: Wave Spectrum

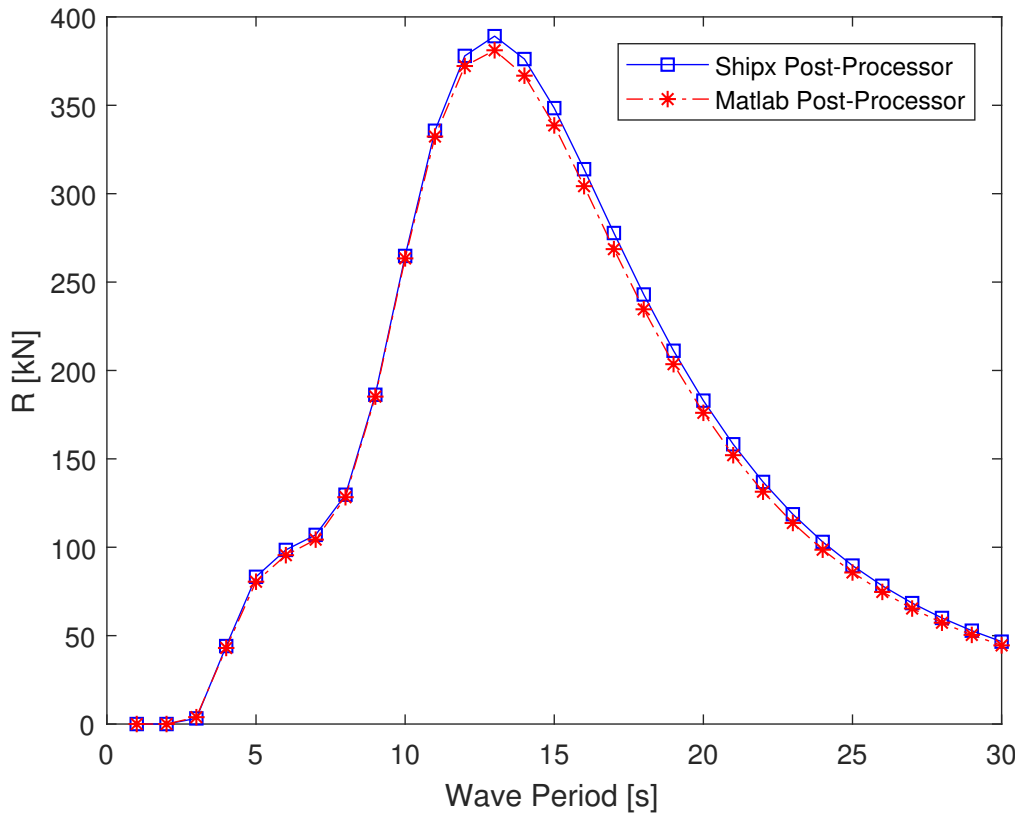


Figure 5.8: Comparison of post-processors for $H_s = 4\text{m}$ and $\beta = 0^\circ$

while 180 degrees hit the ship from behind.

From figure 5.8 we see that the post-processor made for this thesis and the post-processor used in ShipX have close to identical resistance responses as a function of wave period. In this case we have chosen to apply Pierson-Moskowitz wave spectrum to generate the wave spectrum, as this spectrum generated the best fit compared to ShipX's post-processor. This can easily be changed to both Torsethaugen and JONSWAP, depending on operating location and available data. The post-processor function can be found in appendix B.5.1.

5.3.5 Speed Loss Model

From a navigational and hydrodynamic point of view, the accurate calculation or at least reliable estimation of the attainable ship speed at the actual sea is essential from both economic and environmental aspects (Prpic-Orsic et al., 2016).

Speed loss can be divided into voluntary and involuntary losses. Voluntary speed loss occurs when the ship master decides to decrease speed due to for instance navigational hazards, fear

of heavy weather damage from excessive ship motions and accelerations, propeller racing, slamming or boarding seas (Prpić-Oršić and Faltinsen, 2012).

Involuntary speed loss is due to increased resistance from wind and waves. The ship motions and added resistance influences the working point of the propulsion system, which typically increases the fuel consumption and reduces the attainable speed (Prpic-Orsic et al., 2016).

Kwon's Method for Speed Loss Estimation

An approximate method for predicting speed loss of a displacement type ship due to added resistance in waves and wind is presented by Kwon (2008). The method is based on interpretations of detailed calculations of speed loss due to wind and wave reflection. The formula has been shown to provide a good approximation for practical purposes, within a range of block coefficients from 0.55 to 0.85, and Froude numbers from 0.05 to 0.30.

The method requires that the weather conditions are described simply by a Beaufort number (BN) and a relative direction between the ships course and the weather. The Beaufort scale is an empirical measure that relates wind speed to observed conditions at sea or on land. When the Beaufort scale reaches 7 and above, non-linear phenomena like propeller racing are likely to occur. In addition, such conditions are likely to trigger a voluntary speed loss decision from the master of the ship (Kwon, 2008). Therefore, the formula is unlikely to be accurate for Beaufort numbers over 6.

A modification of Kwon's method is presented by Lu et al. (2015), which increases the accuracy of the speed loss prediction. The speed loss is estimated by the following formulae, implemented in a speed loss model block in the hydrodynamic module (figure 5.5).

$$\frac{\Delta V}{V_1} 100\% = C_\beta C_U C_{Form} \quad (5.6)$$

$$V_2 = V_1 - \left(\frac{\Delta V}{V_1} 100\%\right) \frac{1}{100\%} V_1 = V_1 - (C_\beta C_U C_{Form}) \frac{1}{100\%} V_1 \quad (5.7)$$

Formula 5.6 finds the speed loss as a percentage of the original speed, and formula 5.7 shows how to acquire the attainable speed in the given weather conditions. The parameters for these formulae are described in table 5.6. The values of the different parameters are obtained as

Table 5.6: Parameters for Kwon's method, formula 5.6 and 5.7

Symbol	Unit	Description
V_1	[m/s]	Nominal operating speed in still water conditions
V_2	[m/s]	Actual ship speed in the selected weather [m/s]
ΔV	[m/s]	Speed loss, $V_1 - V_2$
C_U	[-]	Speed reduction coefficient as seen in table 5.7
C_β	[-]	Directional reduction coefficient as seen in table 5.8
C_{Form}	[-]	Ship form coefficient as seen in table 5.9

Table 5.7: Speed reduction coefficient C_U by block coefficient (Lu et al., 2015)

Block Coefficient	Loading Condition	Speed Reduction Coefficient C_U
0.55	normal	$1.7 - 1.4F_n - 7.4F_n^2$
0.60	normal	$2.2 - 2.5F_n - 9.7F_n^2$
0.65	normal	$2.6 - 3.7F_n - 11.6F_n^2$
0.70	normal	$3.1 - 5.3F_n - 12.4F_n^2$
0.75	ballast	$2.6 - 1.4F_n - 13.5F_n^2$
0.75	loaded or normal	$2.4 - 10.6F_n - 9.5F_n^2$
0.80	ballast	$3.0 - 1.4F_n - 21.6F_n^2$
0.80	loaded or normal	$2.6 - 13.1F_n - 15.1F_n^2$
0.85	ballast	$3.4 - 1.4F_n + 31.8F_n^2$
0.85	loaded or normal	$3.1 - 18.7F_n + 28.0F_n^2$

shown in tables 5.7, 5.8 and 5.9.

5.4 Machinery Module - Estimation of Total Fuel Consumption

The machinery module reads the output from the hydrodynamics module, and is initiated once the hydrodynamic calculations are complete. The machinery module can be seen in figure 5.9. The purpose of the machinery module is to find the propulsive power needed to overcome the total resistance. The module then sets the engine to produce the required power, calculates fuel consumption and engine loading degree as a percentage of the MCR.

Table 5.8: Direction reduction coefficient C_β due to weather direction (Lu et al., 2015)

Weather Direction	Encounter Angle [deg]	Direction Reduction Coefficient C_β
Head sea and wind	0 - 30	$2 C_\beta = 2$
Bow sea and wind	30 - 60	$2 C_\beta = 1.7 - 0.03((BN - 4)^2)$
Beam sea and wind	60 - 150	$2 C_\beta = 0.9 - 0.06((BN - 6)^2)$
Following sea and wind	150 - 180	$2 C_\beta = 0.4 - 0.03((BN - 8)^2)$

Table 5.9: Ship form coefficient C_{form} due to ship categories and loading condition (Lu et al., 2015)

Type of Ship	Ship form coefficient C_{form}
All ships (except container ships) in loaded condition	$0.5BN + BN^{6.5} / (2.7 \times \nabla^{2/3})$
All ships (except container ships) in ballast condition	$0.7BN + BN^{6.5} / (2.7 \times \nabla^{2/3})$
Container ships in normal loading condition	$0.7BN + BN^{6.5} / (22 \times \nabla^{2/3})$

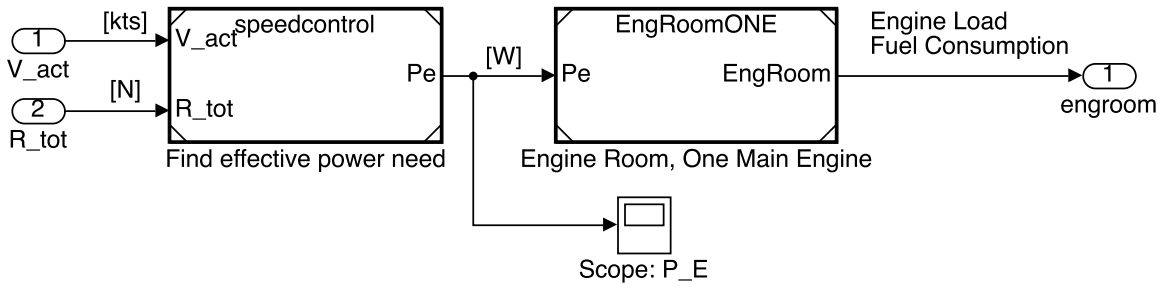


Figure 5.9: Machinery module

5.4.1 Propulsive Power

In order to maintain a steady speed, the ship needs to produce a propulsive thrust force equal to the total ship resistance, described in section 5.3.1. This force is produced by the engines, transferred to the propeller(s), before finally accelerating the water which results in the thrust force. This force transfer is associated with losses, which means that the engine has to produce more power than is effectively used to propell the ship. In the following we will explain how we model this force transfer when starting with the effective power.

The effective (towing) power P_E is the power you would need to tow the ship at the given speed. It is defined as

$$P_E = V_S R_{TS} \quad [kW] \quad (5.8)$$

where V_S is the ship speed and R_{TS} is the total resistance. This is the starting point when we want to estimate the break power P_B required from the engine to maintain speed at the estimated resistance. The effective power is found in the speedcontrol block seen in figure 5.9.

In order to predict the break power that the engine needs to produce in order to reach the effective power, one can use efficiencies. Some of the most important efficiencies and their typical ranges are presented in table 5.10. Note that these values depend heavily on the type of ship and ship design and are only to be taken as average values.

Table 5.10: Efficiencies and their typical values for ships with one propeller (MAN B&W Diesel, 2011; Bertram, 2012)

Symbol	Definition	Name	Range
η_D	P_E/P_D	Propulsive Efficiency	0.40 - 0.70 [-]
η_S	P_D/P_S	Shaft Efficiency	0.96 - 0.995 [-]
η_M	P_D/P_B	Mechanical Efficiency	0.97 - 0.98 [-]
η_O	P_T/P_D	Propeller Open Water Efficiency	0.35 - 0.75 [-]
η_H	P_E/P_T	Hull Efficiency	1.1 - 1.4 [-]
η_R	–	Relative Rotational Efficiency	1.00 - 1.07 [-]

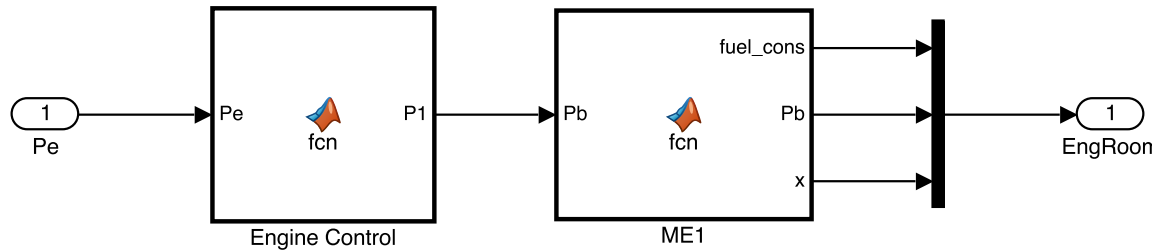


Figure 5.10: Engine model with one main engine

Combining equation 5.8 and table 5.10, we can derive equation 5.9. It is seen that if the propulsive efficiency and the effective power is known, one can obtain the power required to drive the propeller. Furthermore, if the mechanical efficiency is known, the break power requirement for the engine can also be found.

$$P_B = \frac{P_E}{\eta_T} = \frac{P_E}{\eta_M \eta_D} = \frac{P_E}{\eta_M \eta_O \eta_H \eta_R} \quad [kW] \quad (5.9)$$

5.4.2 Engine Model

Once the effective power requirement is found, this is sent to the engine model, seen in figure 5.10. The engine control block then finds break power need following equation 5.9. The engine model can be configured to have between one and four main engines. Depending on the number of engines, the power requirement is distributed so that each engine works as close to its optimal loading degree as possible. The engine model can also be expanded to include a battery.

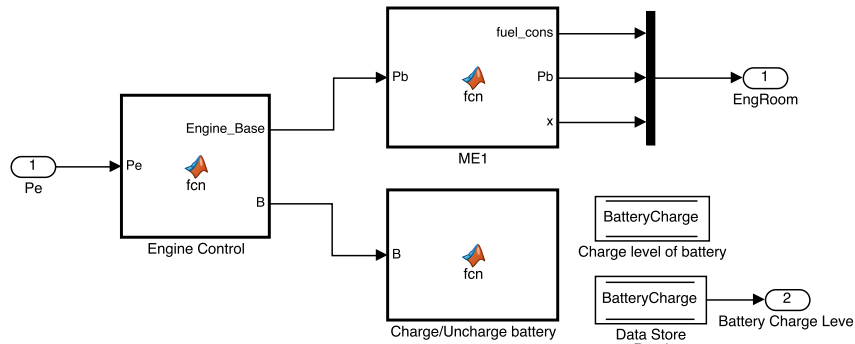


Figure 5.11: Engine model with one main engine and a battery

The engine model requires the user to enter the engine's specific fuel consumption, MCR and optimal loading degree. The output from the engine model is fuel oil consumption and engine loading, both the numerical value, and as a factor of MCR (x). The engine's fuel oil consumption is calculated by the following formula

$$\text{Fuel Consumption} = P_B * \text{SFC} * 1000 \quad [\text{kg/h}] \quad (5.10)$$

5.4.3 Engine Model With Energy Storage System

An engine model that includes an energy storage device, i.e. a battery, can also be included in the machinery module. This model is shown in figure 5.11. Here, the main engine is set to work at a certain load level. This level can be chosen based on how the engine is optimized, in order to minimize fuel consumption.

The engine control block finds the difference between the required break power, and the base power produced by the main engine. If the engine produces more power than is needed, this surplus power is stored in a battery. If the power requirement is higher than what the main engine produces, the battery provides the difference. This means that the engine is always working at its optimal point, and peaks in power demand are handled by the battery. The battery charge level is stored in a data store memory block, seen in figure 5.11. This data memory can be accessed by the Matlab and Simulink function blocks.

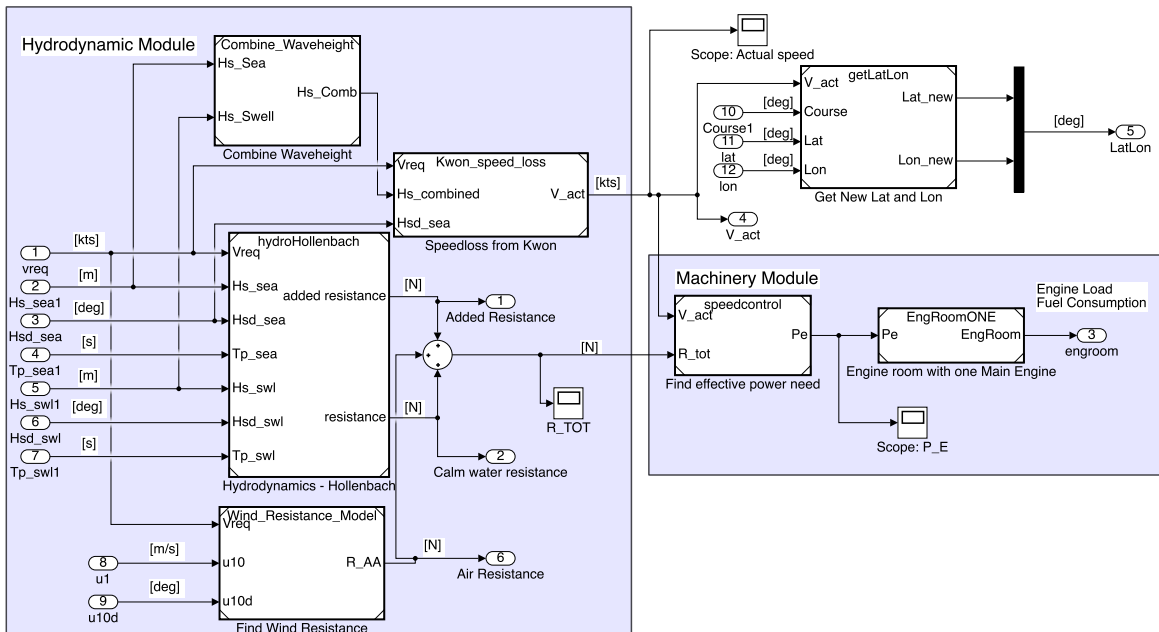


Figure 5.12: Hydrodynamic and machinery modules assembled in the ship model

5.5 Assembling the Modules

Now that the modules have been explained individually, they are assembled in a main model. The machinery and hydrodynamics modules are first assembled in a ship model, as can be seen in figure 5.12. The modules are assembled by connecting the lines from the outputs port of the hydrodynamics module with the input ports from the machinery module.

Within the ship model there is also a block called get new lat and lon. This block is the same for all configurations, and is connected as shown at the top right corner of figure 5.12. It takes in the attainable speed, course, and recent coordinates, and finds the new coordinates. The block uses built in Matlab functions, and the code can be seen in appendix B.2.2

Once the ship model is complete, it is connected to the MetOcean module as shown in figure 5.13. The operations module does not need to be connected by dragging lines, as it uses a function call to connect to the MetOcean module. It does however need to be placed in the same Simulink model. The complete simulation model should now look like figure 5.13.

There are many combinations of models that can be used to form the complete ship model. The machinery module can have between one and four main engines, and one main engine paired with a battery. The hydrodynamics module can use either Hollenbach or ShipX to calculate the

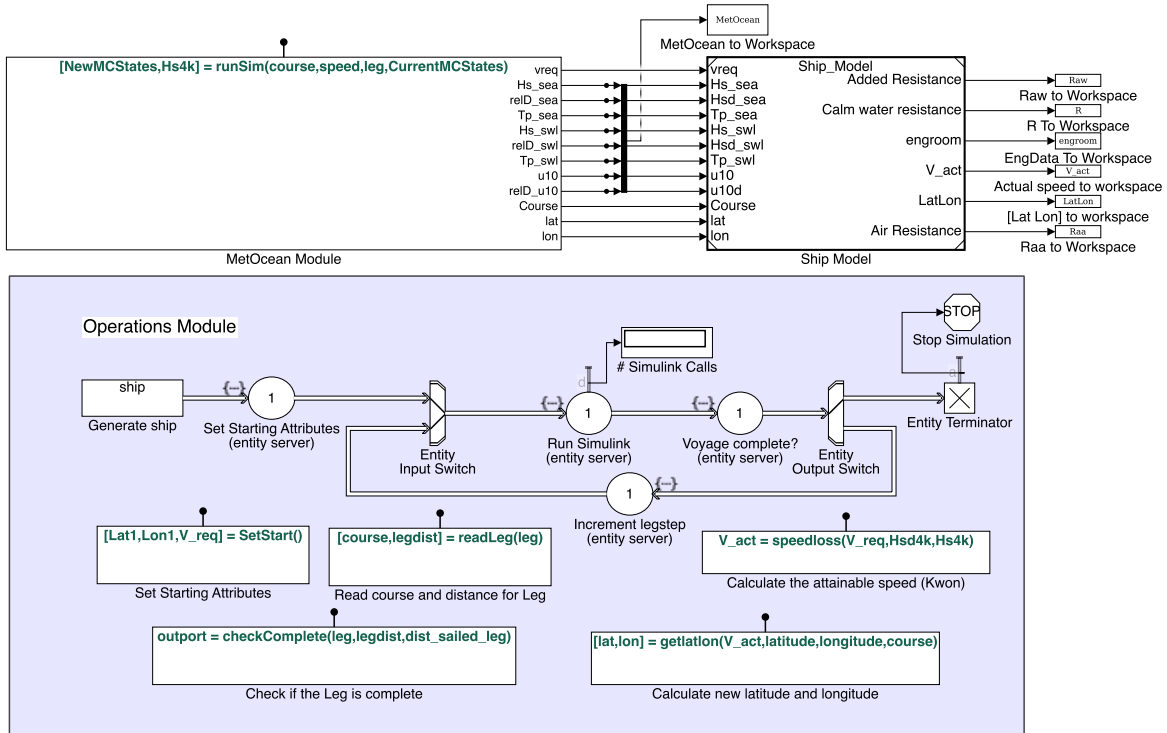


Figure 5.13: Top level view of MetOcean and operations module, and ship model

resistance. All of these possibilities are contained within model blocks, that can be dragged and dropped to configure the ship model as desired by the user. All model blocks developed in the work with this thesis are collected in a block library (found in the electronic appendix).

6. Results and Validation

In this chapter the simulation platform is tested and validated by a case study. The purpose of the case study is to make sure that the model behaves as expected, and produce results agreeing with the physical world. The chapter also includes more extensive testing of the platform, from direct comparison with the case study to the effect hull transformation has on the results.

6.1 Case: Star Lysefjord

In order to validate our simulation model and show how it can be used, a case study was performed. The case is based on data provided by Grieg Star, one of the partners in SFI Smart Maritime. The data is from the open hatch general cargo vessel Star Lysefjord over a period of time from 2014 to 2017. The data is recorded as a time series, with 15 minute intervals. Information about the case vessel Star Lysefjord can be found in table 6.1, and the different data types available can be seen in table 6.2.

6.1.1 Case Description

First the data set is analyzed to find a voyage with high data quality, i.e. no missing entries. By investigating the data records we identified a voyage with high data quality and relatively long duration, from Yancheng in China, to Panama. In table 6.3 we present the details from the voyage. These values are found by summing and averaging entries from the voyage data set.

Table 6.1: Star Lysefjord particulars (Grieg Star, 2013)

Parameter	Description
Type	Open Hatch General Cargo Carrier
Built	2013 Hyundai Ulsan, Korea
LOA	204,4 m
Draught	12,7 m
Breadth Moulded	32,3 m
Design Speed	15,5 kts
Main Engine	One Hyundai B&W 5S60ME-C8
MCR	10780 kW
Deadweight	50 728 t
Gross Tonnage	37 447

Table 6.2: Description of data from Star Lysefjord

Parameter	Description	Unit
Latitude & Longitude	Latitude and longitude of position	[deg]
ME Consumed	Fuel oil consumed by main engine	[t/24h]
ME Load Measured	Main engine load in % of MCR	[-]
Shaft Power	Shaft power	[kW]
Shaft RPM	Shaft speed	[rpm]
Shaft Torque	Shaft Torque	[kNm]
Aux Electrical Power Output	Output of auxiliary engine(s)	[kW]
Draft Fore & Aft	Fore and aft draft	[m]
GPS Speed	Speed over ground	[kts]
Log Speed	Speed through water	[kts]

Table 6.3: Data from case voyage

Parameter	Description
Origin	Yancheng, China
Destination	Panama
Voyage Duration	25 days 1 hour
Voyage Track Length	8800 nm
Average GPS speed	14,63 kts
Average Speed Through Water	14,75 kts
Average Shaft Power	5561 kW
Average Main Engine Load	50,6 %
Total Fuel Consumption	662 t

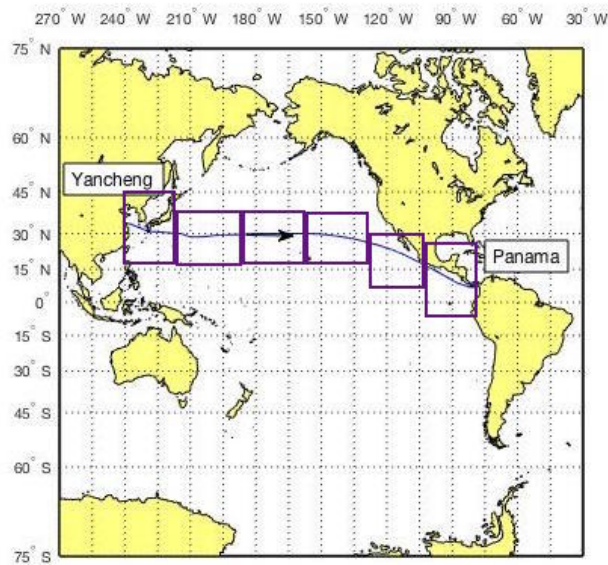


Figure 6.1: Plot of the voyage from Yancheng to Panama. The boxes show where MetOcean data is collected

Figure 6.1 is a plot of the vessel coordinates from the data set, showing the route sailed by the vessel. Next, we use these coordinates to extract hindcast MetOcean data. We extract data from 6 points along the route; one close to each coastline, and four evenly spread along the route. The boxes in figure 6.1 indicates where the MetOcean data is extracted.

The simulation platform is then configured according the case vessel. The ships main particulars are entered into a spreadsheet, containing all the information about the vessel. Engine data is also entered into the spreadsheet, using the information provided by Grieg Star. The vessel has a single main engine and three gensets producing electrical power, and the power produced by these are available. This gives us the possibility to separate propulsive power from hotel consumption and other consumers aboard, as all the power produced by the main engine goes directly to the propulsion. This is beneficial as it allows us to capture the interactions between hydrodynamic resistance and powering need more clearly than if we would have to separate the two ourselves.

6.1.2 Results

The simulation is carried out with both the Hollenbach module (section 5.3.3), and the ShipX module (section 5.3.4) for resistance calculations. In figure 6.2 the break power needed from both simulation with Hollenbach and ShipX are plotted, as well as the actual data from the case voyage. Figure 6.3 shows the percentage deviation between the break power from the data set, and the simulation runs with Hollenbach and ShipX respectively. Furthermore, figure 6.4 shows the significant wave height of the combined sea (Eq. 5.1) encountered in the Hollenbach run. Figure 6.5 shows the direction of this combined significant wave height relative to the ships course, wrapped between 0 and 180 degrees. Figure 6.6 shows significant wave heights encountered in the ShipX simulation, and figure 6.7 the relative directions.

Interpreting Results from the Case

When investigating figure 6.2, it is clearly seen that there are significant differences between the break power from the data set, and the simulation results. The average values, presented in table 6.4, show that using the Hollenbach model gives an average break power need 19% higher than the data set. The ShipX model overestimates the average break power need by 16%.

The most influential cause for error are inaccuracies in calculations for both ShipX and Hollen-

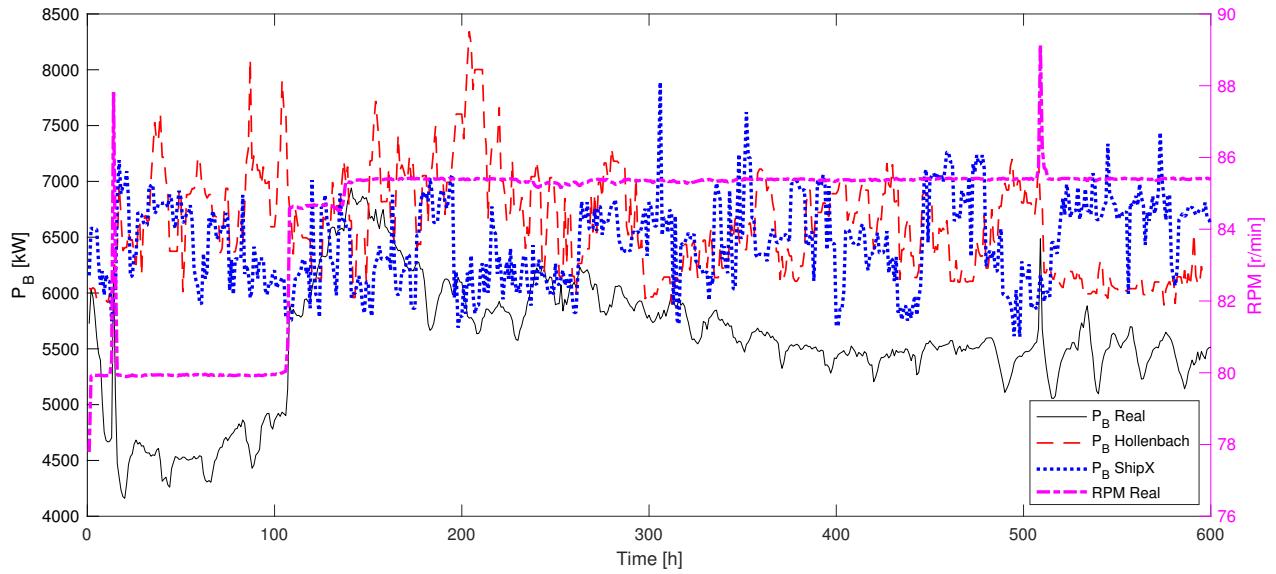


Figure 6.2: Comparison of the break powering need estimated by simulation, and the operational data

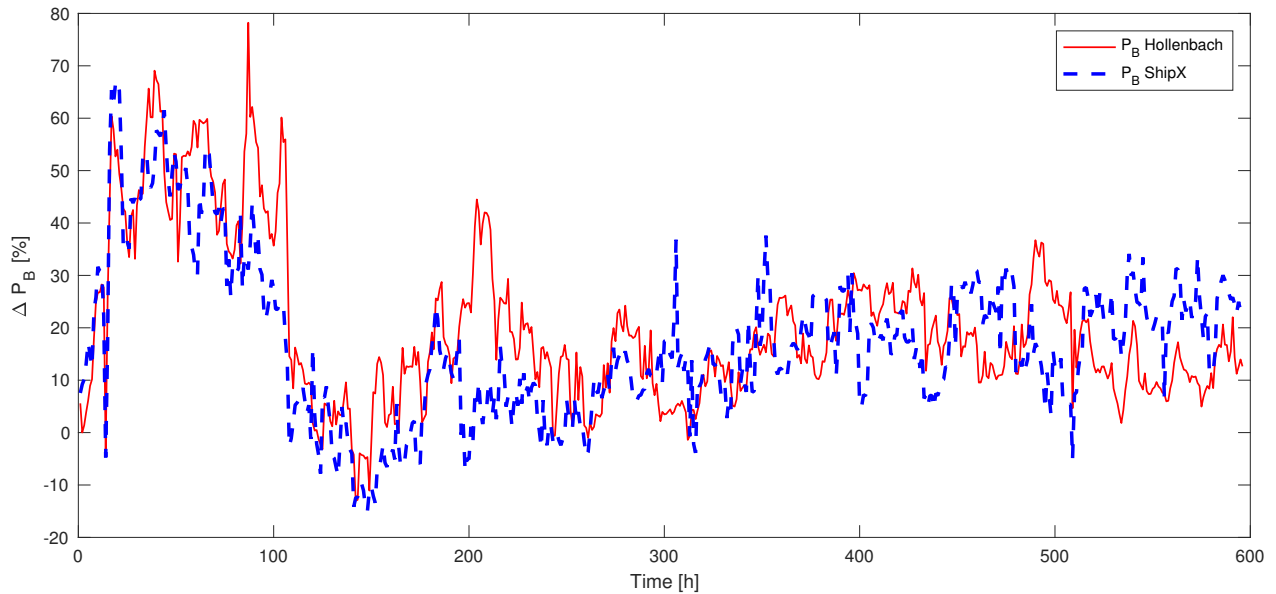


Figure 6.3: Percentage deviation in break power between data set, Hollenbach and ShipX

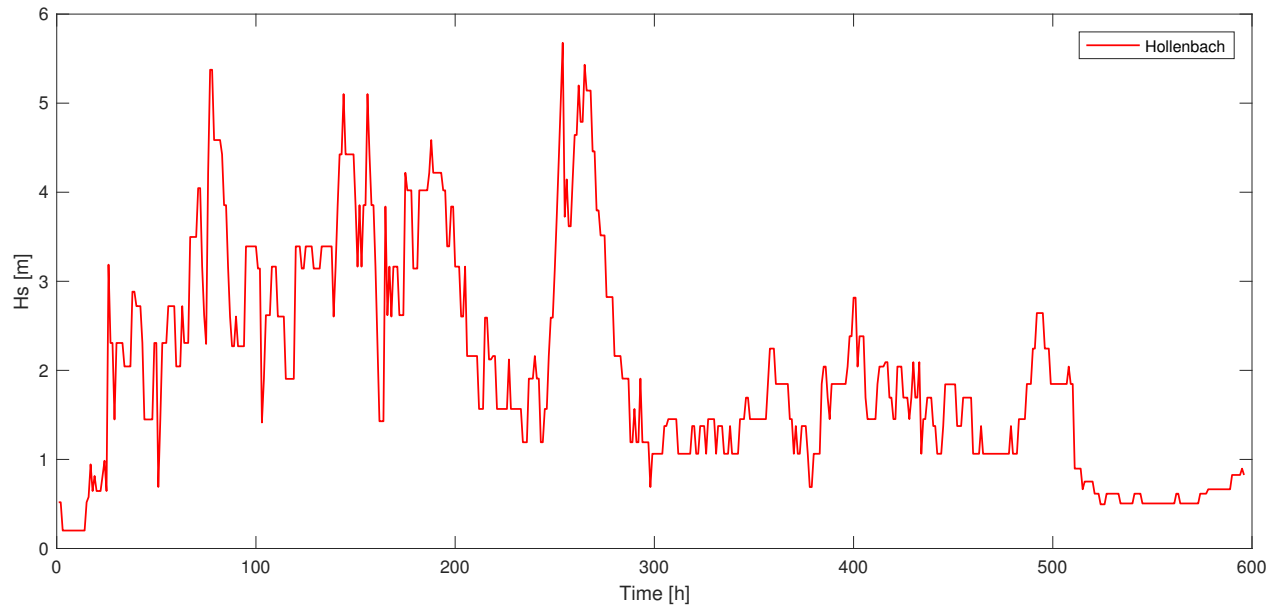


Figure 6.4: Encountered combined wave height for the Hollenbach run

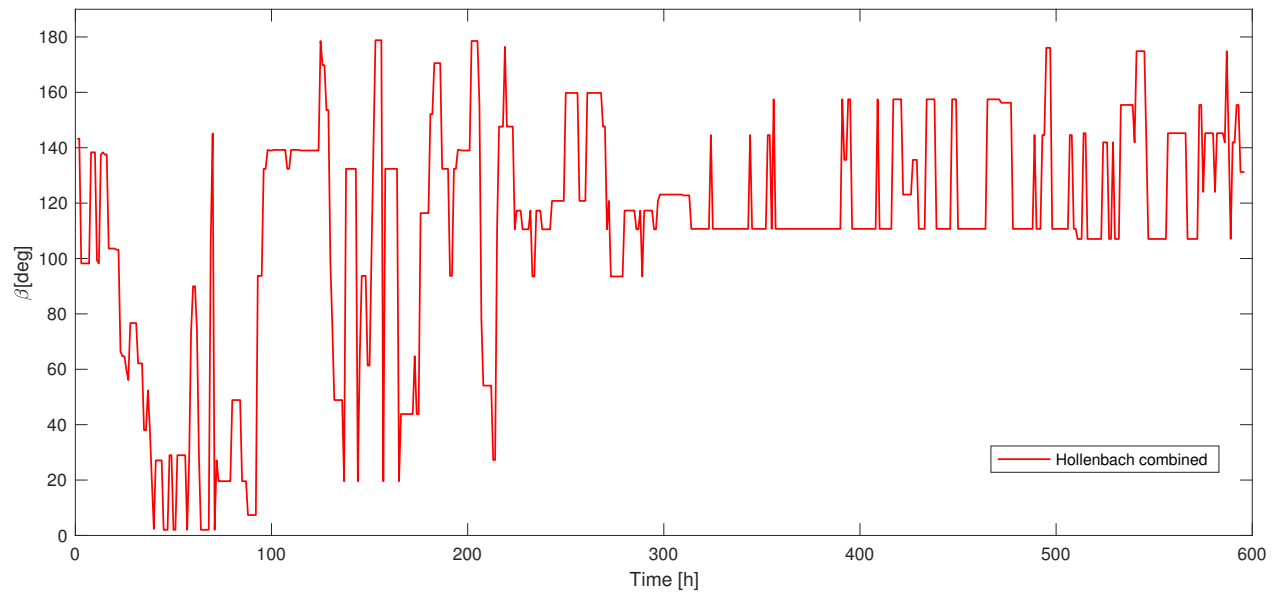


Figure 6.5: Encountered wave direction for the Hollenbach run

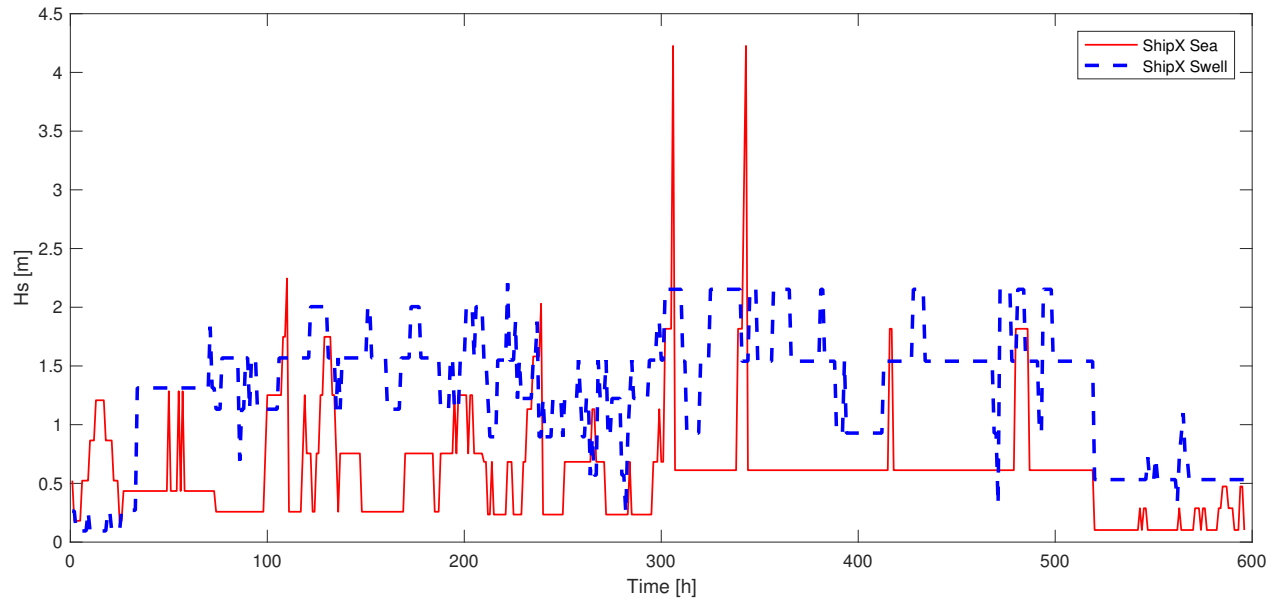


Figure 6.6: Encountered wave heights, sea and swell, for the ShipX run

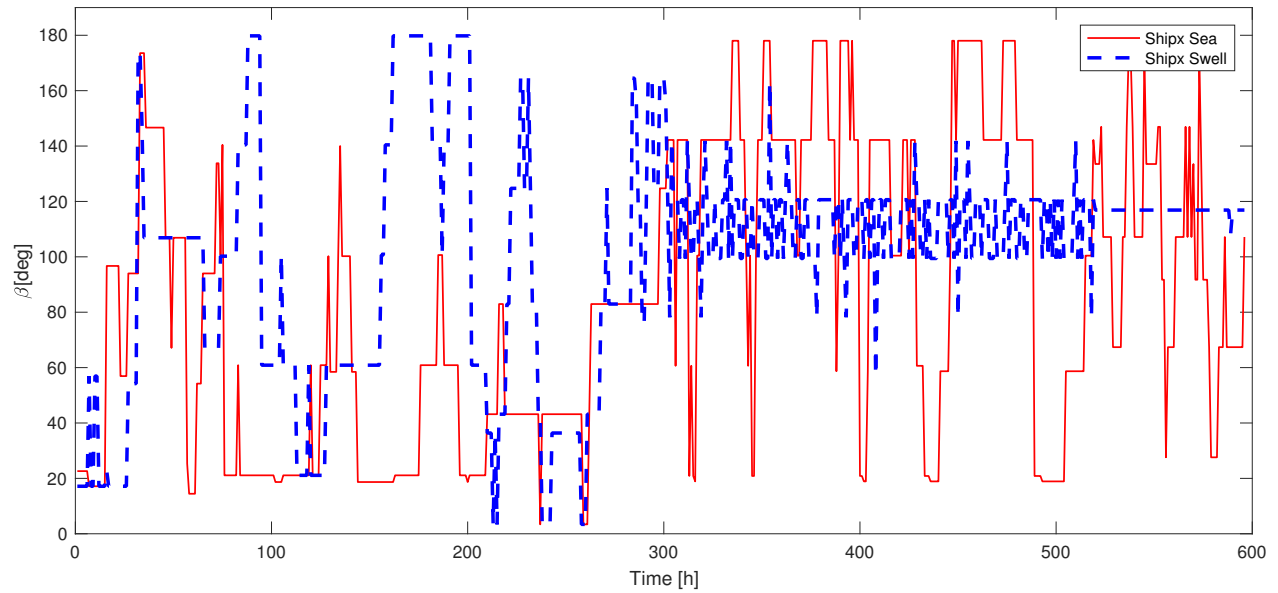


Figure 6.7: Encountered wave direction, sea and swell, for the ShipX run

Table 6.4: Summary of results from case voyage simulation

Parameter	Unit	Real data	Hollenbach	ShipX
L_{pp}	[m]	204.60	204.60	204.60
B	[m]	32.28	32.28	32.28
T	[m]	8.83	8.83	8.83
C_B	[-]	0.72	0.72	0.72
Sailing time	[h]	611	611	611
Avg. V	[kts]	14.75	14.75	14.75
Avg. V Attainable	[kts]	N/A	13.92	14.05
Avg. R_{tot}	[kN]	N/A	647	625
Avg. P_B	[kW]	5561	6610	6456
Fuel Consumption	[ton]	662	699	685
Simulation Time	[s]	N/A	20	≈ 4500

bach. The error margins for both ShipX and Hollenbach are presented and discussed later on in section 7.2. There we show that the percentage increase for calm water resistance calculations for both methods as opposed to the model tests of the case vessel are about 18% while sailing at 15 knots. With a deviation from the case data of 16% it is probable that this is the main cause for the large differences.

Another factor that could contribute towards the differences in average break power, is the that Star Lysefjord did not sail with a constant RPM during the case voyage. The RPM from the voyage is shown in figure 6.2. During the first 100 hours, the RPM was set significantly lower than the rest of the voyage. As can be seen in figure 6.3, it is during these 100 hours that the difference between the actual data and the simulation results, are the largest, reaching almost an 80% difference in the Hollenbach run, and 65% in the ShipX run.

During these first hours there is also a peak in the RPM, coinciding with a peak in the registered break power. This sudden spike indicates that the captain has increased the ship speed, which could be due to navigational or maneuvering needs. Using data from the case voyage, the correlation coefficient between the RPM and the vessels speed through water is found to be 0.83. This indicates, as expected, a strong correlation between these two factors. This means that the decisions taken by the captain has a strong influence on the vessel performance. Because the simulations only include external influences, i.e. MetOcean, this contributes to the differences seen in figure 6.2.

The correlation coefficient between ship GPS speed and ME Load Measured is only 0.52, which only corresponds to a moderate relationship. This indicates that there are external influences on the ship that is not captured in the data (waves). The correlation coefficient is higher when evaluating the ships speed through water, at 0.58. However, these low correlation coefficients

indicate that the ships speed is more related to the propeller RPM than to the break power produced by the main engine.

From figure 6.2 it is also clear that the simulation results have much larger oscillations in break power, than the data set. The simulation results can vary over almost two thousand kW within hours, while the real data typically varies only hundreds of kW. This is a result of the fact that the simulation model is sensitive to MetOcean conditions.

The simulation results are directly dependent on the total efficiency of the propulsion system. In this case, the total efficiency is taken as 0.7. Changing this will not change the form of the curves in figure 6.2, but it changes the average values. This is a potential source for error, as we do not know the actual efficiency of the case ship.

The quality of the data set is of course also a concern. We have no way of verifying the quality of this data set, and there could be other factors that have had an influence on how the actual vessel performed during this voyage, that are not included in the data set. For example, the data from the ship does not contain information about the weather. The weather conditions along the route were simulated for the same time of the year as the actual voyage took place (December 2016 and January 2017). It is however not possible to replicate the exact same conditions.

6.2 Altering trim and dimensions

In this section we have investigated the effects that a change in dimensions for the vessel can have on the performance. In table 6.5 the vessel dimensions used can be seen along with a selection of the results. All of these runs have required a maintained speed of 15 knots through the entire voyage, and they have been exposed to exactly the same weather conditions, i.e. the same seed was used in the random number generator. The three later vessels are all identical to the vessel operating under design conditions except for one parameter, which is written in bold. While vessel one and two only have experienced changes in trim, the beam is increased for vessel three. Compared to the vessel tested in section 6.1 they all have a lot higher resistance due to the over three meter increase in draught. The draught was increased to coincide with the vessels design draught.

The beam of the vessel was increased using the hull transformation tool in ShipX. It is clear that this change increased the vessels required power through the voyage, and consequently the fuel consumption. Other changes can also be done using the transformation tool, such as draught

Table 6.5: Effects that a change of dimension have on the results

Parameter	Unit	Design Cond.	Vessel 1	Vessel 2	Vessel 3
L_{pp}	[m]	204.60	204.60	204.60	204.60
B	[m]	32.28	32.28	32.28	33.28
T	[m]	12.00	12.00	12.00	12.00
C_B	[-]	0.75	0.75	0.75	0.75
S	[m^2]	9514	9459	9560	9684
∇	[m^3]	60,233	59,542	60,030	61,623
Trim, + = fore	[m]	0	1	-1	0
Sailing time	[h]	572	572	572	572
Constant V	[kts]	15	15	15	15
Avg. R_{rot}	[kN]	862	894	842	910
Avg. P_B	[kW]	9460	9784	9322	9962
Fuel Consumption	[ton]	957	992	945	1010
Hydrodynamic mod.	[-]	ShipX	ShipX	ShipX	ShipX

and length.

Altering the trim also resulted in a change in the vessels resistance. It can be seen that by trimming the vessel bow down, the vessel required less power to maintain 15 knots. This difference is probably because of the bulbous bow, as it is more submerged with a trim and more efficient that way. The opposite, trimming the bow up, resulted in an increase in total resistance.

The figures 6.8, 6.9 and 6.10 are the results from one simulation run, with Vessel 2 in table 6.5. From figure 6.8 it is clear that the added resistance can have a big impact on the total resistance if the weather conditions are rough.

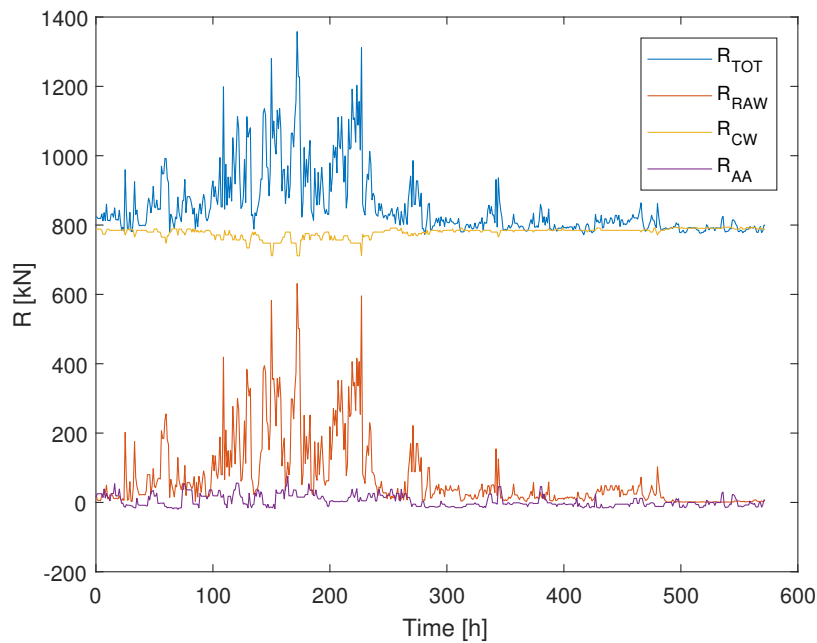


Figure 6.8: Resistance components and total resistance for Vessel 2 during transit

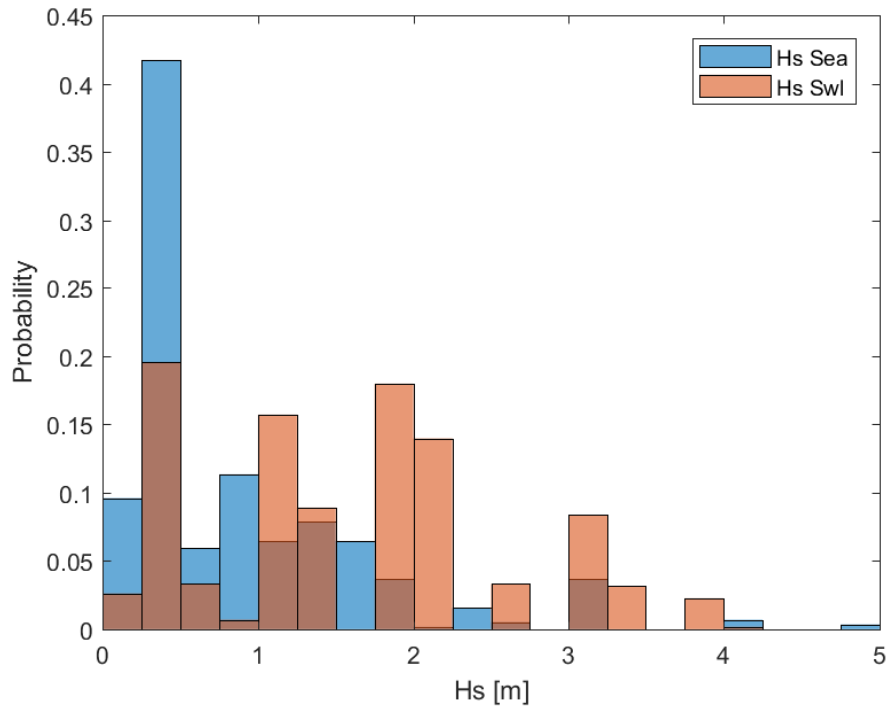


Figure 6.9: Empirical probability distribution for significant wave height period during transit

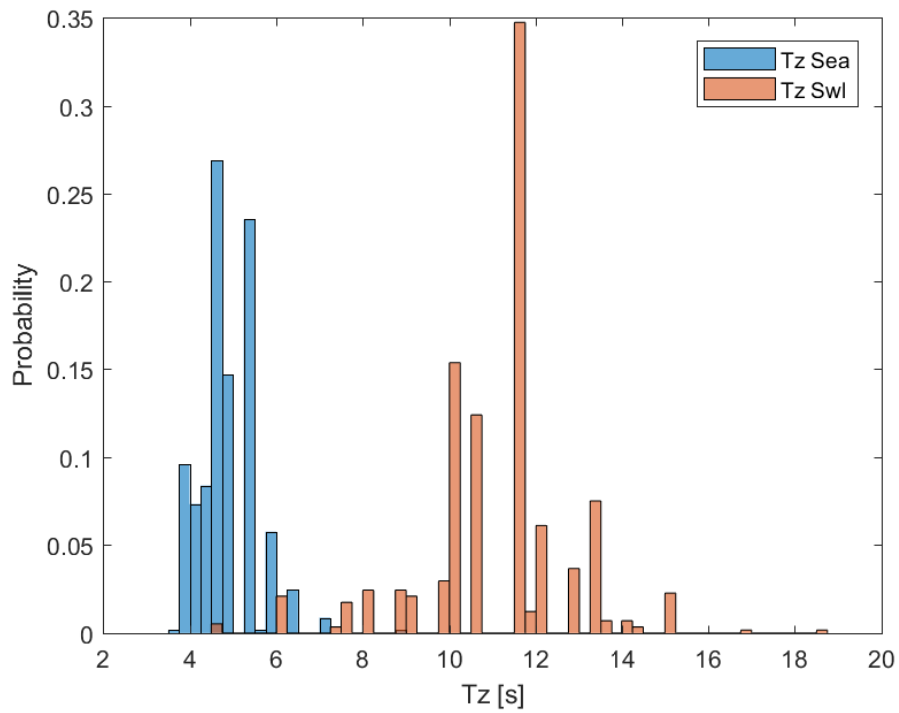


Figure 6.10: Empirical probability distribution for zero-crossing wave period during transit

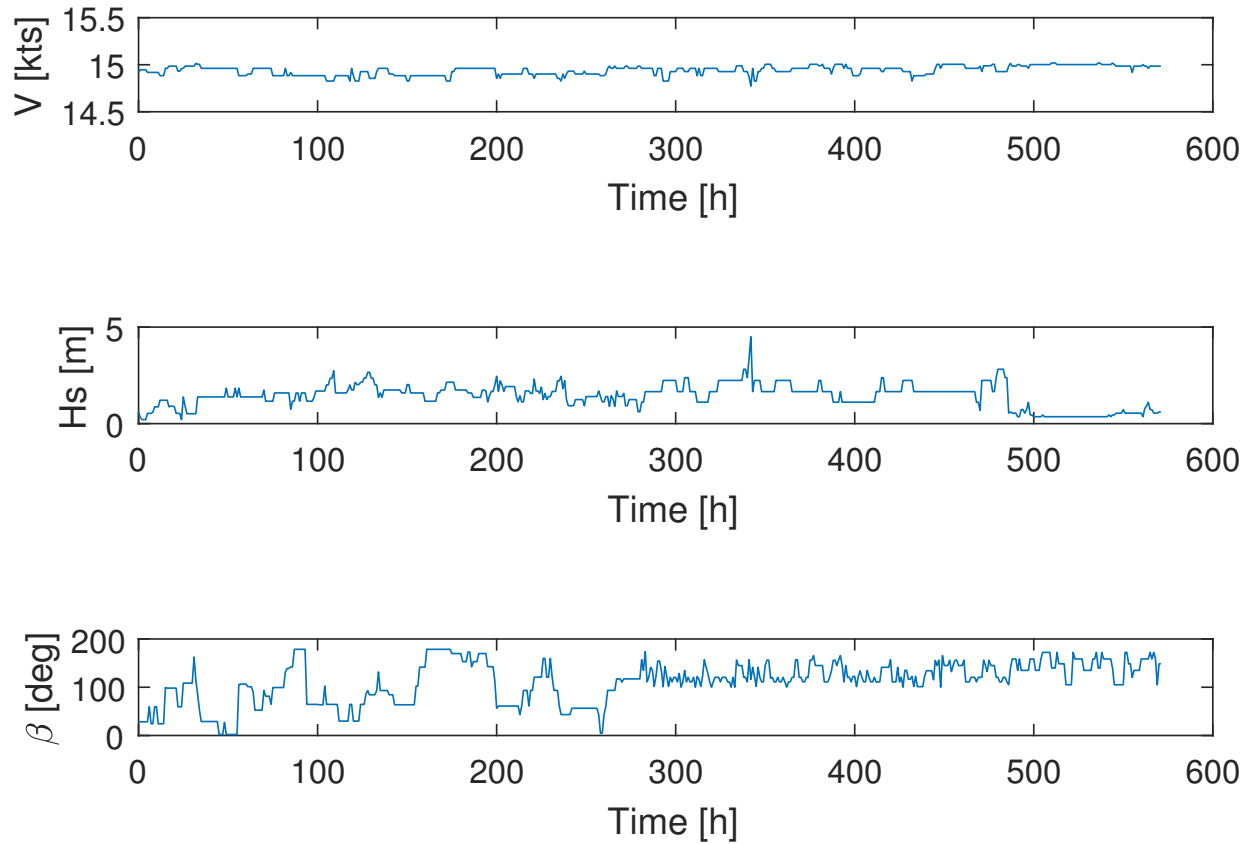


Figure 6.11: Attainable speed with relevant wave height and direction for Vessel 2 during transit

6.3 Varying Speed

As described in sub-section 4.2.2, there are several ways to model speed and power as a function of each other. In figure 6.11 Vessel 2 had a speed requirement of 14.5 knots, with variation in speed according to speed loss calculated using Kwon's method. From the figure we see that the attainable speed is lower when encountering large oncoming waves.

Figure 6.11 shows low fluctuations in maintained speed, this is because the oncoming waves are relatively small. This also leads to small fluctuations in the resistance (6.12) and consequently the break power (6.13). Except for a bigger leap around the 350th hour, the power curve is fluctuating around 9000 kW. The leap up to nearly 12000 kW is a consequence of a large wave crossing from behind. As waves from behind do not make that big of an impact in Kwon's speed loss calculation, the vessel does not reduce the speed that much and has to deal with a quite large resistance, resulting in high power demand.

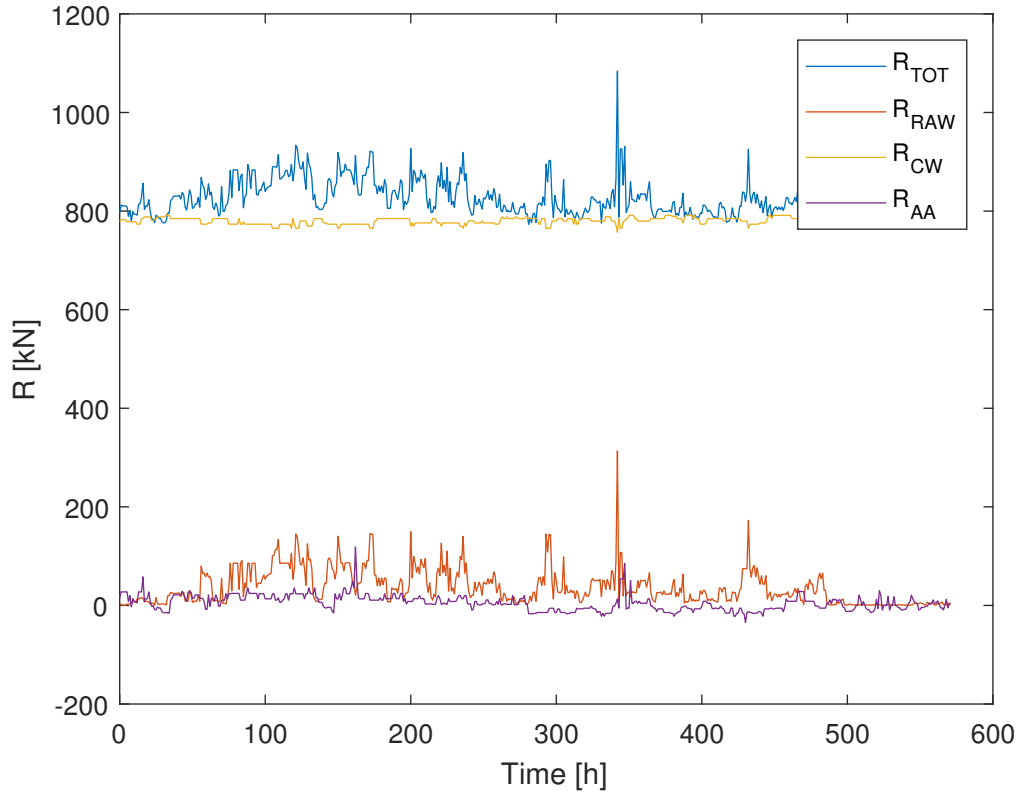


Figure 6.12: Resistance components for Vessel 2 during transit

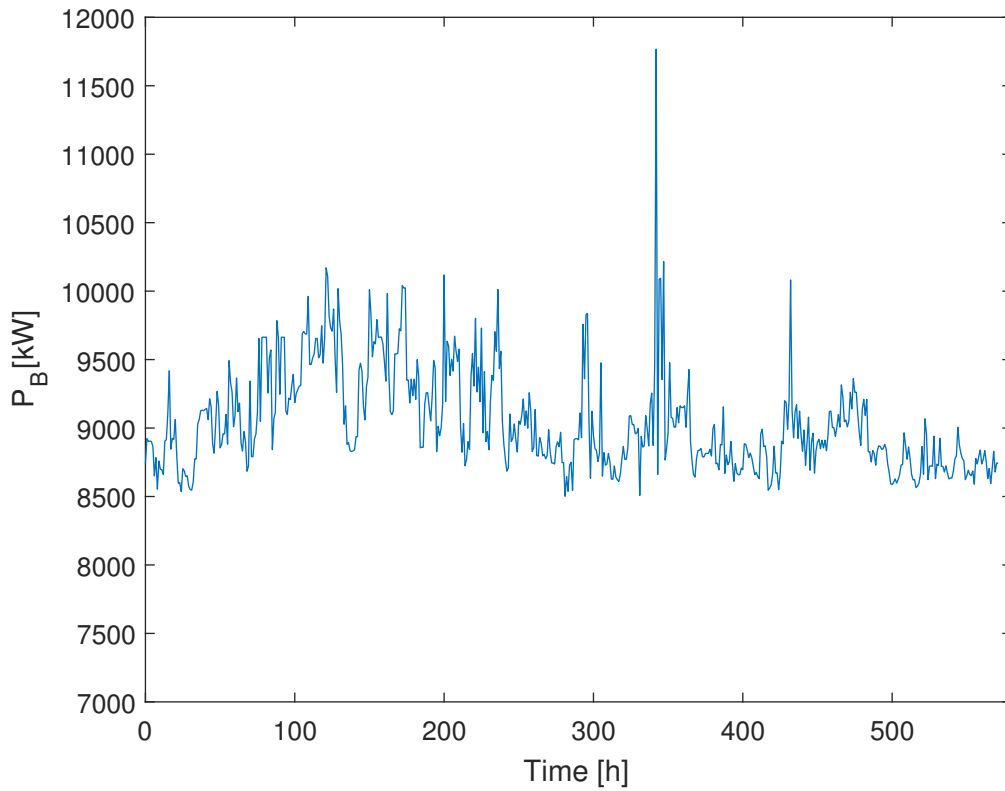


Figure 6.13: Break power for Vessel 2 during transit

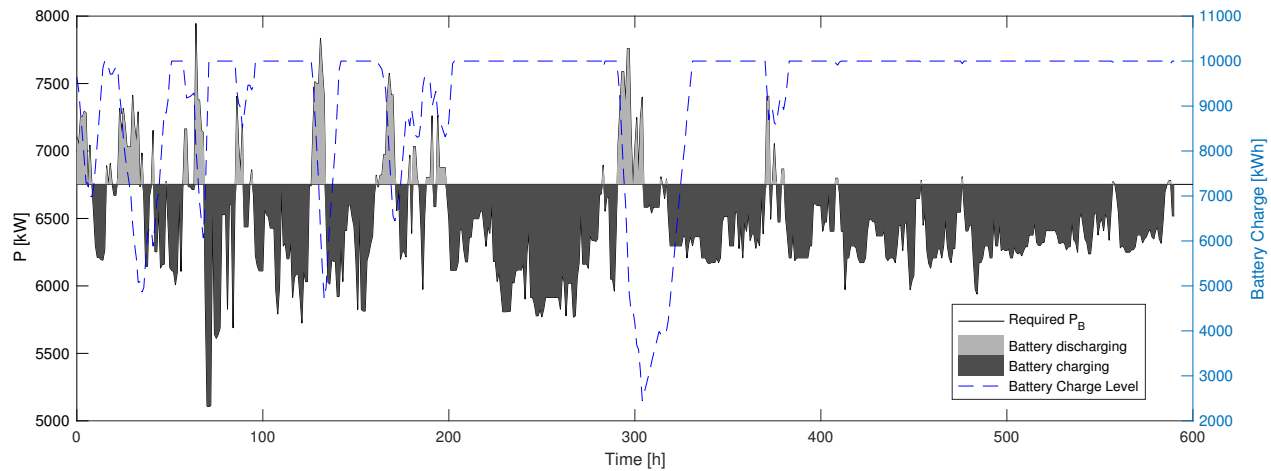


Figure 6.14: Power demand and battery charge level during transit

6.4 Simulating a Ship with Energy Storage System

As described in section 5.4.3, the machinery module can be fitted with an engine and battery model. The battery capacity and its initial charge level can be configured in the battery charge level data store memory block. The engines optimal loading degree is configured in the master script. The benefit of having a battery is that the peaks in power demand can be covered without increasing engine work load. This in turn means that the engine can work at a constant load.

Figure 6.14 shows the results of a simulation run with a battery with a capacity of 10 MWh. The initial charge is also 10 MWh, at the start of the voyage. The ship used in this run is Star Lysefjord, at the same conditions as in the case. The voyage is the same as the case voyage. The engine is set to run at 61.5% of MCR, producing a constant base load of 6752 kW. The speed is set to 14.75 kts, and this speed is used in the resistance calculations, as opposed to using the attainable speed after speed loss. The Hollenbach model is used to calculate the resistance.

Figure 6.14 illustrates a way in which the simulation platform can be used to test an innovative new solution, such as a battery. The use of Hollenbach to calculate the resistance means that the simulation is completed within seconds, allowing for multiple iterations with changes in battery capacity and engine load. As can be seen from figure 6.14, the battery capacity is maxed out for quite a large portion of the voyage. However, the high demand for power around 300 hours almost drains the battery. This is insight that could be used to choose the dimensions of the energy storage system.

7. Discussion

This chapter presents a discussion concerning methods and assumptions applied in this thesis. The aim is to clarify the validity of the model, as well as its shortcomings.

7.1 Method and Application

The process and applied methods for this thesis and for the making of a simulation platform are all based on a desire to maximize the work done by the simulation model itself, limiting the amount of manual work done by the user. It is expected that this can increase the willingness of engineers and designers to implement simulation in their approach to ship design. Also, the illustrative interface of Simulink is making the user aware of the modules interaction, without the need for detailed explanations.

To use the simulation model the user must have a clear goal in mind, and should have some thoughts regarding ship size and dimensions. The general area of operation should also be known, so that accurate weather data relevant to the location is used.

The reduction of necessary work done ahead of simulation is first and foremost noticed when the ShipX module is applied, with Veres running in batch mode. The drawbacks here are as previously mentioned the increase in computational time.

7.2 Resistance Calculation

In the figures 7.1 and 7.2 calm water resistance have been calculated for the case vessel at design conditions. For Hollenbach the deviation is stable around 15% for the entire speed range, which is in agreement with expected deviations for the Hollenbach method. The ShipX results are more alarming, the constant increase in deviation as speed increases can be caused by several factors.

The most likely reason is shortcomings for the model used in ShipX. The model imported to ShipX was based on a 3D-model supplied by Grieg Star. The 3D-model had an irregular form both in the bow and in the aft. Both abeam and for the projections in the water-plane, it showed

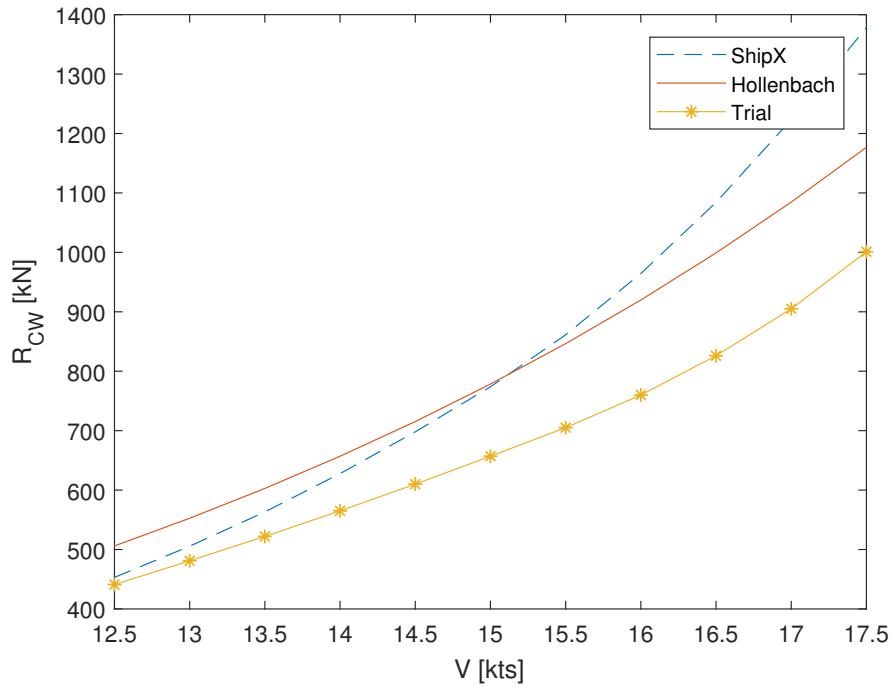


Figure 7.1: Calm water resistance for case vessel at design conditions

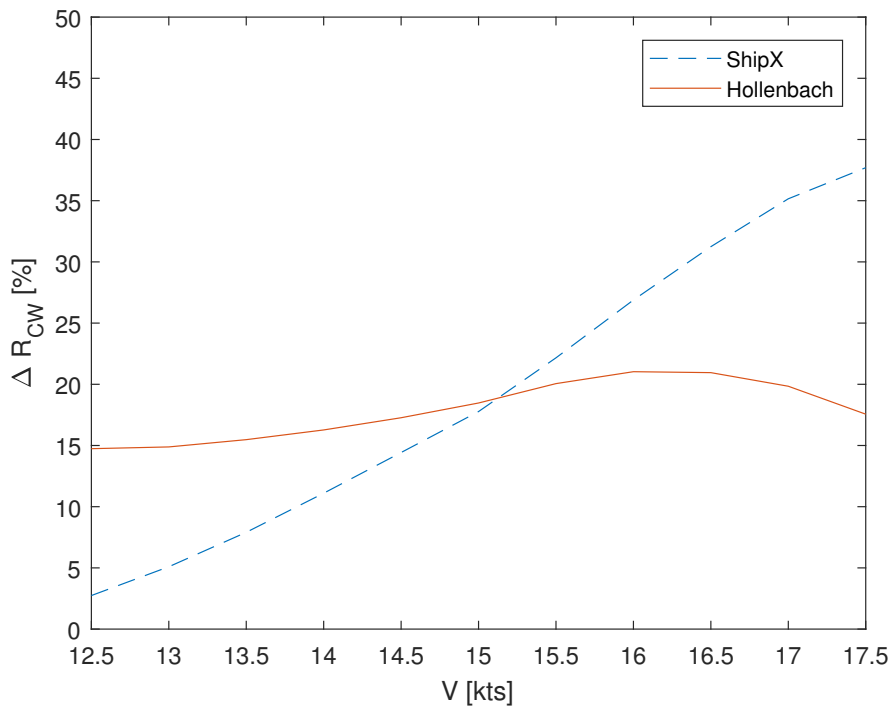


Figure 7.2: Percentage deviation from trial results in figure 7.1

that curvature around the ship was made up of many straight lines, and not curved lines. Another source of error could be how the plug-in *Waveres* in *ShipX* is used, and the calculations performed by *Waveres*. As we only had one vessel with data and model available, it was not possible to validate the *Waveres* calculations with another vessel and case, which should be done.

The *Hollenbach* model is based on empirical methods, and as such it has limited accuracy. In addition, the empirical formula used to calculate added resistance had a valid range only up to two meters waveheight. If the waves were higher, the added resistance was taken as 20% of the calm water resistance. This simplification is not accurate enough to capture the effects of encountered weather. Also, the method did not account for the weather direction. This all contributes towards the mistrust of the added resistance found when using the *Hollenbach* model.

Currents are not included in the model. Currents could influence the flow around the hull, and alter the speed seen by the hull and propeller. This could influence the ship resistance, and leaving it out is a source of error. It is however not believed to have a large influence on the total resistance.

7.3 Computational Time

A short computational time is not necessarily a priority, but we have seen that simulating one transit while applying *ShipX* has taken more time than expected. This is due to the need for generating wave spectres. A change that can cut the run-time for the model by close to 50% is to apply the *Torsethaugen* spectrum as opposed to the *Pierson-Moskowiz*, and then use combined-sea values for the weather conditions. Other than that it is of course possible to use *Hollenbach* as the hydrodynamic module, which will reduce the computational time from about 4500 seconds to 20 seconds (6.4).

Other ways to decrease computational time, but increase the workload for the user before the model can be initialized, is to make large tables for the added resistance with weather conditions and vessel speed as variables. This would take a lot of work in *ShipX* using plug-ins like *Veres* or *Ship Speed and Powering*.

7.4 Modeling MetOcean Conditions

One alternative approach for application of weather data could be to use historical records for a set of years, and then run several simulations with the same interval of years, but a different period. These data could then be applied directly. Analyzing the output from these simulation could give us an indication towards the deviations over the years, and what one could expect. Alternatively, weather forecasts could be used. This could be demanding in terms of computational time, but it would open the door towards weather routing in future work. Sampling weather conditions using separate Markov chains for each leg of the voyage as well as for each of the different wave characteristics requires insight from the user. First of all it requires that the user acquires historical weather data, and that the data is representative for the entire voyage. Should the user only choose to acquire weather data representing a location near the shore the data would consequently not be representative for a leg in the middle of the Atlantic. Therefore it is imperative that the user chooses as many legs covering the voyage as is necessary to ensure correct representation of the conditions along the route. The necessary number may have to be found by trial and error, where the encountered conditions should be compared to long term wave statistics in the same area.

An assumption that has been made in the module where Markov chains are used, is that the weather conditions; wave height, direction and period are independent of each other. Ideally one would take into account all weather conditions, and also the correlation between each of the parameters. That would most likely increase the run-time for the model, while there would be little contribution to the models accuracy. According to [Pantuso et al. \(2015\)](#) the correlation between random variables have very little influence on the final value when modeling a stochastic phenomena. This study was only validated for their specific case, but gives an indication to the insignificance of correlation in stochastic modeling. Also, the correlation between the parameters could prove to be negligible. Figure 7.3 shows the joint correlation for all legs in the case presented earlier (section 6.1). The highest correlation is 0.48, between wave direction and mean period. This corresponds to a moderately low correlation. The calculated correlation values are for a small data set and might not be valid for other journeys.

The added benefit of using Markov chains are that they will ensure a more realistic development for the weather as the ship sails, as it is based on transitions between states and probability of that transition. Another approach could have been to create a distribution (normal or otherwise), and draw conditions from that. This would have made the average of the conditions more

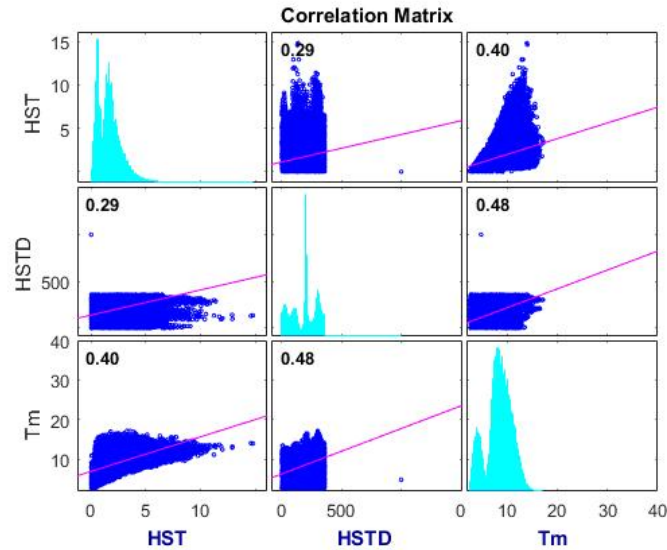


Figure 7.3: Correlation between weather parameters

or less similar to that of the Markov chain, but the consistency throughout the trip would not be as realistic.

An example where the consistency of weather conditions are important would be during a dynamic positioning operation where the operation is depending on an acceptable weather window. The procedure in such a case could be to simulate a forecast for the weather, while checking to see if there is a big enough window with acceptable weather conditions. If so, the ship would start to operate. An option could also be to have a probability for inaccurate forecasts, meaning that the weather window could be shorter than that which was found in the forecast.

7.5 Estimating Power Requirement

7.5.1 Power as a Function of Speed and Resistance

In this thesis, the power requirement has been found as a function of ship speed and resistance. In reality, many ships are RPM driven, meaning that the captain controls the thrust force by increasing or decreasing the engine speed. This means that the power produced by the engine is not directly related to the resistance. If the ship encounters a large wave set, the ship would slow down due to added resistance from the waves unless the captain increases the engine speed. On the other hand the ship speed would increase when conditions go from rough

to calm waters without altering the engine speed. This could be described as constant power, varying speed.

The attainable speed, resulting from the speed loss calculations, can also be used to estimate the propulsive power need and resistance calculations. In this case, the model will estimate a lower power need, because the power depends on the speed and resistance. This approach means that the ship only has to produce as much power as is needed to keep the attainable speed. In other words, the model predicts a speed loss which is never really experienced - because it only produces enough power to overcome the resistance at the reduced speed.

The output from the machinery module is heavily dependent on what efficiencies are entered by the user. This can be beneficial at an early stage, as the efficiencies can typically be found by looking at similar ships and propulsion systems. This means that the platform can be configured rapidly and with limited research. The drawback is that you are not able to increase the accuracy of the output to such a degree that the output could be trusted.

7.5.2 Auxiliary and Hotel Loads

Auxiliary and hotel loads are left out of the machinery module at this stage. Several ship types, like cruise and OSV, have significant amounts of these loads compared to propulsive power. However, we chose to model the operations of a vessel as if hotel and auxiliary loads are not influenced; assuming that the power demand of these consumers are not influenced by external conditions such as waves and wind. The user will have to take this into account, and the estimated power needed to supply other consumers than the propulsion system should be added to the power production capability. This could be done by including separate gensets, or increasing main engine capacity.

7.5.3 Hull Fouling and Degradation

No degrading factors are included in the simulation platform. This means that we model the ship as if it did not degrade during its operational life. Over time, organisms will grow on the wet part of the ships hull. This will disturb the flow around the hull, and cause an increase in frictional resistance. The propeller could also experience degradation, reducing its efficiency. Normally, this is dealt with by the ship owners by periodically docking the ship, re-painting the hull and removing organisms from the propeller.

The degradation of ship hull and propeller could have relatively large impacts on ship performance, especially over longer time periods. This effect is not captured by the simulation platform.

7.6 Accuracy of the Model

As far as accuracy for the model is concerned, there is clearly room for improvement. As previously mentioned the calm water resistance is incorrect compared to trial runs of a physical model. The calculations should be validated for other ship models to see if it is a repeating error, or if it is a specific error for the tested model. Disregarding the calm water resistance issue, we find that the model responds as expected to weather changes and vessel changes.

8. Conclusion and Further Work

8.1 Conclusion

Based on a study of the state of the art within simulation, a simulation platform applicable in early stage ship design has been built. The platform is able to simulate a vessels performance during sailing, while exposed to stochastically modeled weather conditions based on historical data in a given area. The work can be concluded as follows:

- The model catches and illustrates important interactions between a ships sub-systems as well as external factors.
- The module based modeling approach makes it quick and easy to perform iterations with design changes in between. This improves the decision making process by providing the designer with a tool for rapid design testing and performance evaluation.
- The differences between the hydrodynamic modules allows for testing of a ship as the ship design process moves along, from Hollenbach with few necessary inputs, to ShipX where the need for input is greater. However, relatively small differences were found, especially considering the increased computational time.
- The MetOcean module simulates realistic weather conditions based on historical data. Inclusion of Markov chains ensures memory in the model, making sure there are natural transitions between weather states.
- Applying wave spectres for both swell and wind-sea to calculate added resistance gave accurate results compared to using a combined spectrum.
- When evaluating and comparing vessel changes, the model has produced satisfactory results as to which design performs better. It is possible to see the effect one change has on the entire system.
- We find that added resistance can amount to a large part of the total resistance, and should be among the deciding factors for ships that are set to operate in rough conditions.

8.2 Further Work

In this section we will highlight some of the possibilities that have emerged from the simulation platform. Throughout the work with this thesis we have identified many areas where our simulation platform could have an impact. At the same time we note that the platform should be thoroughly tested and improved, according to the discussion in chapter 7, before it is applied towards the ideas described in this section.

Ship Route Planning - Weather Routing

Ship route planning and weather routing are operational measures that can be applied to minimize emissions and reduce sailing time. In future work, the simulation platform developed in this thesis could be used towards this purpose. The platform could be used to test different routes, and evaluate the vessel performance and attainable speed for each route. This could be implemented in a cost function, and the optimal route could then be chosen based on the calculated cost. Should the platform be used for weather routing however, it should be expanded to include weather forecasts, and not only hindcast data. Because the platform is already capable of using MetOcean data from different locations, this is deemed an interesting possibility. This implementation of weather routing could essentially make the platform a real-time decision making tool, where missions could be distributed between a fleet so that the fleet performance is optimized.

Including Optimization

Simulation is used in this thesis as a means to decrease lead-time and improve the accuracy of ship performance evaluation, with the hope of an improved final design. In future work the platform developed in this thesis could be applied towards optimization. One approach to this could be to transform the output from the simulations into a cost-function, where some parameters are chosen to be used as optimization variables. From this cost function, a response surface could be created, in which an optimal point could be found.

Optimizing a ship design is a multi-objective optimization problem with a complex structure, where change in parameters could greatly affect the outcome. Because the result of these changes can be captured in our simulation platform, an optimization model would not have to involve

extensive work towards creating a cost function. It seems to be manageable to create such an optimization model, and is something that could be considered in future work.

Reduce Computational Time

A programming property that could be included to reduce computational time when the user wants to run the same ship for a longer period of time, is to include a memory in the model. This implies that during simulation the model remembers vessel responses to given weather conditions, so that if they should occur again, the result is already known. For sailing the same route over a period representing 20 years for example, this would be relevant.

Machinery

Normal procedures for the power management system, regardless of the composition aboard a vessel, is to have all running engines run on the same load. Introducing optimal running loads based on power need can make the final engine layout different than that of a optimization run where all engines run on the same load. This of course requires that conservative chiefs aboard the vessel are prepared to change their routines and the producers of power management systems are ready to implement this change. There are of course safety requirements that must be taken into account when writing such an algorithm.

Using FMI and more complex black box models

In the future, the platform should be expanded with models/modules of already developed components, like engine models etc. The platform should follow the standardized interface FMI, in order to ensure the correct communication between the models. This could improve the accuracy of the model when a high fidelity level is required, and not necessarily reduce the user-friendliness. Because of a well defined interface, it should be easy to implement these models.

Including Ship Motions

There are more ship responses that should be included in addition to added resistance, to better capture the performance of the design. Operability based on ship motions and limits for these is

an interesting parameter. For example if one wants to include dynamic positioning as part of the operational profile of a ship tested in the model. In light of how the platform is able to capture realistic operating conditions, this expansion seems doable. More DOF's should be included, in order to more accurately find the responses of the ship in waves.

Include Propeller Model

In order to increase the accuracy of the simulation platform, the machinery module should be expanded to include a propeller model. The thrust force produced by the propeller will vary with the water inflow speed, which is affected by the ships wake, and other factors that are currently not included in the platform. The propeller model should include all relevant factors, evaluated at the current conditions (ship speed, wave conditions etc.), and then convert the power delivered to the propeller to a thrust force. This thrust force should then be compared to the total resistance force.

This would effectively allow for propeller optimization, where the actual operating profile forms the basis for the calculations.

Calculate EEDI and Benchmark Designs

In future work, the platform could be configured to calculate the EEDI of ships. This would allow for a benchmark value that could be used to compare the performance of the design to other ships. Other benchmarking alternatives should also be explored. Measuring the overall performance of a ship is no simple task, but the platform developed in this thesis provides a good foundation on which a benchmarking methodology could be built.

Expanding the Operation Module

In future work, the round trip operation module should be expanded to include cargo modeling. The port operations can be expanded to include limitations such as crane availability. By adding queuing in ports, modeling cargo availability and loading/unloading, the round trip model can give realistic estimates of the annual cargo capacity of the vessel. Because the platform is able to estimate the attainable speed, good estimations for annual amount of cargo transported could be obtained.

Bibliography

- Amdahl, J., Endal, A., Fuglerud, G., Hultgreen, L., Minsaas, K., Rasmussen, M., and Vallan, H. (2005). TMR 4100-Marin Teknikk Intro, TMR4105 Marin Teknikk 1.
- Banks, C., Turan, O., Incecik, A., Theotokatos, G., Izkan, S., Shewell, C., and Tian, X. (2013). Understanding Ship Operating Profiles with an Aim to Improve Energy Efficient Ship Operations. In *Low Carbon Shipping Conference*, pages 1–11.
- Banks, J., Carson, J. S., Nelson, B. L., and Nicol, D. M. (2000). *Discrete event system simulation*. Englewood Cliffs, 3rd edition.
- Bassam, A. M., Phillips, A. B., Turnock, S. R., and Wilson, P. A. (2016). Design, modelling and simulation of a hybrid fuel cell propulsion system for a domestic ferry.
- Becker, M. C., Salvatore, P., and Zirpoli, F. (2005). The impact of virtual simulation tools on problem-solving and new product development organization. *Research Policy*, 34(9):1305 – 1321.
- Bertram, V. (2012). *Practical ship hydrodynamics*. Elsevier.
- Blockwitz, T., Otter, M., Akesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., and Viel, A. (2012). Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. pages 173–184.
- BMT ARGOSS (2017). Wave Climate. <http://www.waveclimate.com>. [Online; accessed June-2017].
- Bø, T., Johansen, T. A., Dahl, A. R., Miyazaki, M. R., Pedersen, E., Rokseth, B., Skjetne, R., Sørensen, A. J., Thorat, L., Utne, I. B., Yum, K. K., and Mathiesen, E. (2015). Real-time Marine Vessel and Power Plant Simulation. In *Proceedings of the ASME 2015 34th International Conference on Ocean, Offshore and Engineering*, volume 1, pages 1–10.
- Erikstad, S. O. (2015). Design processes objectives and requirements. Lecture Note TMR4115 Design Methods.
- Erikstad, S. O., Grimstad, A., Johnsen, T., and Borgen, H. (2015). VISTA (Virtual sea trial by simulating complex marine operations): Assessing vessel operability at the design stage. 1:107–123.
- Evans, J. H. (1959). Basic design concepts. *Journal of the American Society for Naval Engineers*, 71(4):671–678.
- Faltinsen, O. (1993). *Sea loads on ships and offshore structures*, volume 1. Cambridge university press.

- Fathi, D., Grimstad, A., Johnsen, T., Nowak, M., and Stålhane, M. (2013). Integrated Decision Support Approach for Ship Design. *Journal of the American Society for Naval Engineers*, 136(4):671–678.
- Fathi, D. and Hoff, J. R. (2017). SHIPX Vessel Responses (VERES) Ship Motions and Global Loads Users' Manual Software Programvare Hydrodynamics Hydrodynamikk. *Norwegian Marine Technology Research Institute*.
- Fathi, D., Ringen, E., Alterskjær, A., and Berget, K. (2012). ShipX Ship Speed & Powering Plug-In.
- Fathi, D. and Steen, S. (2000). ShipX Wave Resistance Plug-In (Waveres) 3.0-User's Manual Norwegian Marine Technology Research Institute Hydrodynamics Hydrodynamikk Software Programvare.
- Gaspar, H. M., Ross, A. M., Rhodes, D. H., and Erikstad, S. O. (2012). Handling Complexity Aspects in Conceptual Ship Design.
- Grieg Star (2013). Star Lysefjord. <https://www.griegstar.com/vessels/our-open-hatch-fleet/star-lysefjord/>. [Online; accessed May-2017].
- Hassani, V., Rindarøy, M., Kyllingstad, L. T., Nielsen, J. B., Sadjina, S. S., Skjong, S., Fathi, D., Johnsen, T., Æsøy, V., and Pedersen, E. (2016a). Virtual prototyping of maritime systems and operations. In *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*, pages V007T06A018–V007T06A018. American Society of Mechanical Engineers.
- Hassani, V., Simon Sadjina NTNU SeverinSadjina, S., Alesund, N., og Romsdal, M., Stian Skjong NTNU StianSkjong, N., Trondheim, N., and Dariusz Fathi, N. (2016b). Virtual Prototyping of Maritime Systems and Operations (ViProMa).
- IMO (2016). Ship Energy Efficiency Regulations and Related Guidelines.
- ITTC (2005). Full Scale Measurements, Speed and Power Trials, Analysis of Speed/Power Trial Data (7.5-04 \n-01-01.2). *ITTC - Recommended \nProcedures and Guidelines*.
- Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.
- Kwon, Y. (2008). Speed loss due to added resistance in wind and waves. *Nav Archit* 3:14-16, pages 16–19.
- Lu, R., Turan, O., Boulougouris, E., Banks, C., and Incecik, A. (2015). A semi-empirical ship operational performance prediction model for voyage optimization towards energy efficient shipping. *Ocean Engineering*.
- MAN B&W Diesel (2011). Basic Principles of Ship Propulsion.

- MarineWiki (2011). Ship Design Process. http://www.marinewiki.org/index.php?title=SHIP_DESIGN. [Online; accessed May-2017].
- MathWorks (2017a). *Mapping Toolbox™ User's Guide*. The MathWorks Inc. https://www.mathworks.com/help/pdf_doc/map/map_ug.pdf.
- MathWorks (2017b). *Simulink® User's Guide*. The MathWorks Inc. https://www.mathworks.com/help/pdf_doc/simulink/sl_using.pdf.
- Myrhaug, D. (2007). *Kompendium TMR4180 Marin Dynamikk – Uregelmessig sjø*. Akademika Forlag.
- Pantuso, G., Fagerholt, K., and Wallace, S. W. (2015). Which uncertainty is important in multistage stochastic programmes? A case from maritime transportation. *IMA Journal of Management Mathematics*, (January 2015):1–13.
- Pettersen, B. (2007). *Marin Teknikk 3, Hydrodynamikk*, volume 1. Akademika Forlag.
- Prpić-Oršić, J. and Faltinsen, O. M. (2012). Estimation of ship speed loss and associated CO₂ emissions in a seaway. *Ocean Engineering*, 44:1–10.
- Prpic-Orsic, J., Vettor, R., Faltinsen, O. M., and Guedes Soares, C. (2016). The influence of route choice and operating conditions on fuel consumption and CO₂ emission of ships. *Journal of Marine Science and Technology (Japan)*.
- Sadjina, S., Skjong, S., Pedersen, E., Æsøy, V., Kyllingstad, L. T., Rindarøy, M., Fathi, D. E., Hassani, V., Johnsen, T., and Bremnes Nielsen, J. (2017). Distributed Co-Simulation of Maritime Systems and Operations.
- Sánchez, P. J. (2007). Fundamentals of simulation modeling. In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, pages 54–62. IEEE Press.
- Smart Maritime (2017). Newsletter May 2017. <http://www.smartmaritime.no/news/newsletter-may-2017/>. [Online; accessed June-2017].
- Steen, S. and Minsaas, K. (2013). TMR4220 Naval Hydrodynamics Ship Resistance.
- Taskar, B., Yum, K. K., Steen, S., and Pedersen, E. (2016). The effect of waves on engine-propeller dynamics and propulsion performance of ships. *Ocean Engineering*, 122:262–277.
- Thomke, S. H. (1998). Simulation, learning and ramp;d performance: Evidence from automotive development. *Research Policy*, 27(1):55 – 74.
- Yum, K. K. (2016). Simulation of a Hybrid Marine Propulsion System in Waves.

A. Electronic Appendix

Together with this thesis a zip-file was uploaded, containing the following files:

- Master exhibition poster
- Simulation Platform that the thesis is based on. Including Matlab source codes and input files for Veres.

B. Coding

B.1 Main Script used to run Simulation

```
% Main script running the simulation and processing output
%
% Author: Peter Tenfjord and Martin Bakke
% Date: 01.06.2017

clear; close all; clc;

addpath('Blocks')
addpath('Data_Source')
addpath('Functions')
addpath('MetOcean')
addpath('Pre_Processing')
addpath('ReiseSim')

%% Preprocessing
% Read the vessel input file and make the parameters available for the
% Simulink model. All blocks that want to access these data: Data
↳ Editor-->Input-->Parameter

filename = 'Vessel_Info.xlsx';
xlRange = sprintf('B%d:BC%d',5,5); %The numbers at the end specify rows in the file, set
↳ them according to what info you want to use
[num,txt] = xlsread(filename,xlRange);

% Read engine data into EngDat input file
% Configure the engine_pre function according to your system (Functions
% folder)
EngDat=engine_pre();

RoundTrip=0; % set to 1 if simulating roundtrip, 0 otherwise
SetNumRTs=1; % set the number of roundtrips you want to simulate, set to 1 if
↳ RoundTrip=0;

seed=2; %choose the seed to control random number generation
```

```

Random_Values=generate_random_matrix(seed,SetNumRTs); %generate random number matrix

%% Still water Resistance table for shipx, waveres
%Uncomment the next line if you are using ShipX in the hydrodynamic module
%Resistance_Table = waveRes ();

%% Generate markov chains (MC) for weather parameters
N=6; % Number of legs. Configure this according to route (Get Route function)
states = 9; % Wanted number of states for markov chain
FirstMonth = 11; % First chosen month for weather data
LastMonth = 12; % Last month for weather data

% Preallocation of matrices; This is done in all sub-codes, as Simulink
% requires it in order to run.
% MC... matrices are markov matrices ( states x states x leg number)
% ...Range is the value range for the respective period and leg chosen
MC_Hs_sea = zeros(states,states,N); Hs_seaRange = zeros(states,N);
MC_Hsd_sea = zeros(states,states,N); Hsd_seaRange = zeros(states,N);
MC_Tp_sea = zeros(states,states,N); Tp_seaRange = zeros(states,N);
MC_Hs_swl = zeros(states,states,N); Hs_swlRange = zeros(states,N);
MC_Hsd_swl = zeros(states,states,N); Hsd_swlRange = zeros(states,N);
MC_Tp_swl = zeros(states,states,N); Tp_swlRange = zeros(states,N);
MC_u10 = zeros(states,states,N); u10_Range = zeros(states,N);
MC_u10d = zeros(states,states,N); u10d_Range = zeros(states,N);

for j=1:N

    str=sprintf('Leg%d.csv',j);

    % Wind-Sea values
    [MC_Hs_sea(:,:,j),Hs_seaRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,12);
    [MC_Hsd_sea(:,:,j),Hsd_seaRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,13);
    [MC_Tp_sea(:,:,j),Tp_seaRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,14);

    % Swell values
    [MC_Hs_swl(:,:,j),Hs_swlRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,17);
    [MC_Hsd_swl(:,:,j),Hsd_swlRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,18);
    [MC_Tp_swl(:,:,j),Tp_swlRange(:,j)] = MC_param(str,states,FirstMonth,LastMonth,19);

    % Wind values
    [MC_u10(:,:,j),u10_Range(:,j)] = MC_param(str,states,FirstMonth,LastMonth,5);

```



```

    [MC_u10d(:, :, j), u10d_Range(:, j)] = MC_param(str, states, FirstMonth, LastMonth, 6);

end

clear str j N
%% Prepare [course, distnm]
%Run the get_route function to automatically attain waypoints (wpts)
%get_route(origin, destination, ship_speed)

%Or manually input waypoints below
wpts= [];
    %The following waypoints can be used as a guide, they follow a route
    %from China to Panama
%      [32.8502 121.2880;
%      33.5522 128.4185;
%      41.2058 138.4851;
%      46.9253 -179.1512;
%      41.8339 -150.2097;
%      15.3330 -99.8767;
%      7.1122 -79.7435];

% If waypoints are inputted manually, run the following
% to store the course and distance of the legs, to be used in simulation
[course_legs, distnm_legs] = legs(wpts, 'rh');

%If using a roundtrip model, uncomment the following

%if RoundTrip==true
%wptsBA=flip(wpts);
%[course_legsBA, distnm_legsBA]=legs(wptsBA, 'rh');
%end

%% Set the required, or intended, speed for the ship

v_req_start=14.5; %[kts]

%% Execute the simulation

if RoundTrip==true
    tic
    sim('Main_RT.slx');

```

```

    toc
else
    tic
sim('Main.slx');
    toc
end

%% Process the output from simulation
% This data is from the "To Workspace" blocks in the model
t = tout; % Time steps of events
Hs_sea = MetOcean.data(:,1); % Significant wave height, Sea based [m]
relD_sea = MetOcean.data(:,2); % Direction of wave, Sea based [deg]
Tp_sea = MetOcean.data(:,3); % Peak period of wave, Sea based [s]
Hs_swl = MetOcean.data(:,4); % Significant wave height, Swell based [m]
relD_swl = MetOcean.data(:,5); % Direction of wave, Swell based [deg]
Tp_swl = MetOcean.data(:,6); % Peak period of wave, Swell based [s]
u10 = MetOcean.data(:,7); % Wind Speed [m/s]
relD_u10 = MetOcean.data(:,8); % Relative direction of wind [deg]
Raw = Raw.data; % Added resistance [N]
Res = R.data; % Calm water resistance [N]
Raa = Raa.data; % Air resistance [N]
Rtot = Res+Raw+Raa; % Total resistance [N]
Raw_Res = (Raw./Res)*100; % Added resistance as a percentage of calm water
    ~ resistance [%]
ME1FC = engroom.data(:,1); % Main Engine fuel consumption [kg/h]
ME1PB = engroom.data(:,2); % Main Engine Break Power Production [W]
ME1X = engroom.data(:,3); % Main Engine Loading degree [%]
V_att = V_act.data; % The actual attainable speed, from Kwon [kts]
LatPts = LatLon.data(:,1); % The Simulated Latitudes
LonPts = LatLon.data(:,2); % The Simulated Longitudes

%Gather output data in tabular form
if RoundTrip==true
VoyStep=voyage_step.data; % Ship status (in port or sailing) during simulation
%Uncomment the following if there are different lengths in vectors to be gathered in
    ~ table
%VoyStep=[VoyStep;2];
T=table(t,VoyStep,V_att,Hs_sea,relD_sea,Tp_sea,Hs_swl,relD_swl,Tp_swl,u10,relD_u10,Raw,R-
    ~ es,Raa,Rtot,Raw_Res,ME1FC,ME1PB,ME1X);
else

```

```
T=table(t,V_att,Hs_sea,relD_sea,Tp_sea,Hs_swl,relD_swl,Tp_swl,u10,relD_u10,Raw,Res,Raa,R-
    ← tot,Raw_Res,ME1FC,ME1PB,ME1X);
end
```

```
%% Calculate averages for different parameters
```

```
if RoundTrip==true
    R_total = zeros(1,SetNumRTs*2);
    PB_total = zeros(1,SetNumRTs*2);
    Fuel_Cons_Total = zeros(1,SetNumRTs*2);
    SailTimes = zeros(1,SetNumRTs*2);
    num_steps = zeros(1,SetNumRTs*2);
    j = 1;
for j = 1:SetNumRTs*2
    for k = 1:numel(t)
        if VoyStep(k) == j
            num_steps(j) = num_steps(j)+1;
            SailTimes(j) = SailTimes(j)+1;
            R_total(j) = R_total(j)+Rtot(k);
            PB_total(j) = PB_total(j)+ME1PB(k);
            Fuel_Cons_Total(j) = Fuel_Cons_Total(j) + ME1FC(k);
        end
    end
end
```

```
Avg_Res=zeros(1,SetNumRTs*2);
Avg_PB=zeros(1,SetNumRTs*2);
```

```
for i=1:SetNumRTs*2
    Avg_Res(i)=R_total(i)/num_steps(i);
    Avg_PB(i)=PB_total(i)/num_steps(i);
end
```

```
% Create summary table
```

```
SailTimes=SailTimes';
Avg_Res=Avg_Res';
Avg_PB=Avg_PB';
Fuel_Cons_Total=Fuel_Cons_Total';
```

```
Summary=table(SailTimes,Avg_Res,Avg_PB,Fuel_Cons_Total)
else
```

```
Tot_Sail_Time=numel(t);           % [h], as each time step represents 1 hour
SailTimes=Tot_Sail_Time;
Avg_Res=mean(Rtot);              % Average total resistance [N]
Avg_PB=mean(ME1PB);             % average load on Main Engine [W]
Fuel_Cons_Total=sum(ME1FC)/1000; % Total fuel consumption [t]
% Create summary table
Summary=table(SailTimes,Avg_Res,Avg_PB,Fuel_Cons_Total)
end
```

```
%% Plot Simulation results, Ship performance, and Voyage coordinates
```

```
Plot_RunMaster;
```

B.2 Route Interaction

B.2.1 Route Generation

```

%% Function that determines the sailing route
%
% Author: Peter Tenfjord and Martin Bakke
% Date: 01.06.2017

function [distnm_legs,course_legs] = get_route(origin, destination,ship_speed)

% The input of this script should be a starting point, and a destination,
% in addition to ship speed. The output is a set of waypoints along rhumb
% lines that approximate a great circle.
% source: https://www.mathworks.com/help/map/navigation.html

% Starting point [Lat,Lon]
%origin = [33.698882, -118.289108]; %Los Angeles
%origin = [33.384153, 120.830253]; %Yancheng
% Destination [Lat, Lon]
%destination = [35.181525, 140.432739]; %Tokyo
%destination = [8.831395, -79.451721]; %Panama

% draw a great circle between the points
gcpts = track2('gc',origin(1),origin(2),destination(1),destination(2));

% draw the rhumb line between the points
rhpts = track2('rh',origin(1),origin(2),destination(1),destination(2));

% draw N number of waypoints to approximate the great circle
% Use vessel speed and great circle length to calculate the number of waypoints needed
- with 3 hour intervals
dgc = distance('gc',origin,destination);
total_time=distgcnm/ship_speed; % Estimate the time spent on the journey [hours]
distgcnm = deg2nm(dgc); % great circle distance in nm (shortest route)

```

```

N=floor(total_time/8); % calculate the number of legs needed

%N=3;
[latpts,lonpts]=gcwaypts(origin(1),origin(2),destination(1),destination(2), N);

% Compute headings and distances for the waypoint legs, and
[course_legs, distnm_legs] = legs(latpts,lonpts,'rh');

%% plot the voyage
figure('color','w');
ha = axesm('mapproj','mercator','maplatlim',[-75 75],'maplonlim',[90 -30]);
axis off, gridm on, framem on;
setm(ha,'MLineLocation',15,'PLineLocation',15);
mlabel on, plabel on;
%load coastlines;
hg = geoshow('landareas.shp');
    %geoshow(coastlat,coastlon,'displaytype','line','color','b');

geoshow(latpts,lonpts,'displaytype','line')
geoshow(gcpts(:,1),gcpts(:,2),'DisplayType','line',...
        'color','red','linestyle','--')
geoshow(rhpts(:,1),rhpts(:,2),'DisplayType','line',...
        'color','green','linestyle','-.')

end

```

B.2.2 Determining position of vessel at time, h

```
%% Function that finds new coordinates based on speed and course
%
% Author: Peter Tenfjord and Martin Bakke
% Date: 01.06.2017

function [lat_new, lon_new] = New_Lat_Lon(V_act, Course, Lat, Lon)

coder.extrinsic('nm2deg')
coder.extrinsic('reckon')

time=1; % one hour time steps

lat_new=0; %preallocate
lon_new=0; %preallocate

distance=V_act*time; %distance sailed [nm]
arc_length=nm2deg(distance); %distance sailed [deg]

[lat_new, lon_new]=reckon(Lat,Lon,arc_length,Course); %new lat and lon

end
```

B.3 Markov chain

B.3.1 Generation of Markov chains

```

%% Base script to generate a Markov chain transition matrix for a given parameter
%
% Author:   Martin Bakke and Peter Tenfjord
% Date:    01.05.2017

function [MC, StateValue]=MC_param(CSVfile,numStates,firstMonth,lastMonth, col)

% CSVfile - MetOcean data in a csv file.

% numStates - the desired number of states, if this number proves to yield
% an absorbing state the code will reduce number of states by one and try
% again. The resulting matrix will be of the same size as numStates
% requires, but with some rows and cols equal to zero should the code be
% forced to reduce number of valid states.

% firstmonth/lastmonth - Choose valid months, 1 = january, firstmonth must
% be less or equal to lastmonth, including only dec. and jan. will not
% work.

% Col, choose which parameter the transition matrix should be generated for.

MC = zeros(numStates);
StateValue = zeros(1,numStates);

%% Making sure there are no absorbing states in the final matrix
absStates = 2;
while absStates >= 1 && numStates > 0

% Reads the csv file from waveclimate.com into a matrix
A = csvread(CSVfile,23,0);

% Loop over all the rows in the matrix
for i = 1:size(A,1)
%   Check if the month is in the range wanted
    if (A(i,2) >= firstMonth) && (A(i,2) <= lastMonth)
        if A(i,col) > 0 && A(i,col) < 999 % Disregard zero and negative values

```



```

    if (exist('Param') == 0) % Param doesn't exist
        Param = A(i,col);
    else
        % Add the value at the end of the chosen parameter
        % Sets the parameter equal to Param + the additional row which is the
        % value from A(i,col)
        Param = [ Param; A(i,col) ];
    end
end
end
end

% Set number of states in the markov chain
%
% Beware of setting this too high. If there are too many states, some of
% the states will be absorbing, that is,  $P(j,j) = 1$ , which means it can
% never transition to other states.

% Find upper limit for parameter values and divide the values into even bins
ul = max(Param);
umin = min(Param);
% Find state ranges - first state [0,stateRange] and so on
stateRange = ul / numStates;
% Make vector with real size corresponding to state
BigRange = linspace(umin,ul,2*numStates+1);
k=1;
Range = zeros(1,numStates);
for i=2:2:2*numStates+1
    Range(k)=BigRange(i);
    k=k+1;
end
% State values - stateRange, 2xstateRange and so on up til ul
stateValues = stateRange:stateRange:ul;
% Initialize 1D-matrix holding the state of each data point
ParamState = zeros(length(Param),1);

% Find each data points state
for i = 1:length(Param)
    % For each data point
    for j = 1:numStates
        % For each state
        if Param(i) <= stateValues(j)

```

```

        % Data point is in state j
        ParamState(i) = j;
        % This data point is categorized, so we break and move to the
        % next data point
        break;
    end
end
end

% Find transitions
transitions = zeros(numStates);
for t = 1:length(ParamState)-1
    % ParamState(t) represents the state and ParamState(t+1) represents the state
    % it transitions to
    transitions(ParamState(t),ParamState(t+1)) =
-   transitions(ParamState(t),ParamState(t+1)) + 1;
end

P = transitions;
% Normalize each row in the transition matrix so each row sums to 1
for i = 1:numStates
    P(i,:) = P(i,:) / sum( P(i,:) );
end

% Check to see if there are any absorbing states
% i.e.  $P(i,j) == 1$  where  $i=j$ 
absorbstate = zeros(numStates);
for i = 1:numStates
    for j = 1:numStates
        if P(i,j) == 1
            absorbstate(i,j) = absorbstate(i,j) + 1;
        end
    end
end

absStates = sum(sum(absorbstate));
numStates = numStates-1;

end

if numStates == 0
    P = 1;

```

```
    Range = mean(Param);  
end  
  
% Making all matrices the size of numstates x numstates  
% Easier to work with when making 3D matrices  
  
[A, B] = NaNtoZero(P,Range);  
  
MC(1:length(A),1:length(A)) = A;  
StateValue(1:length(B)) = B;  
  
end  
% Transition matrix is now ready
```

B.3.2 Next state is found

```

%% Function that finds the next state in the Markov chain based on the previous one.
%
% Author:  Martin Bakke and Peter Tenfjord
% Date:    01.05.2017

function [Sea_HSS, Sea_HSDS, Sea_TPS] = Get_Metoccean (Hs_SeS, Hsd_SeS, Tp_SeS, MC_Hs_sea,
    ~ MC_Hsd_sea, MC_Tp_sea, leg, Random_Values)
% Hs... - Current state
% MC... - Markov chain for that parameter
% leg - Which leg of the voyage the vessel is on
% Random_Values - Pre-generated matrix with random numbers
% Sea... - Next state

persistent randnum

if isempty(randnum)
    randnum=1;
end

r=0;
Sea_HSS = 0;
Sea_HSDS = 0;
Sea_TPS = 0;

%% Run through for each of the separate Markov Chains
for three_mat=1:3
    if three_mat==1
        P = MC_Hs_sea;
        state = Hs_SeS;
        col=1;
    elseif three_mat==2
        P = MC_Hsd_sea;
        state = Hsd_SeS;
        col=2;
    else
        P= MC_Tp_sea;
        state = Tp_SeS;
        col=3;
    end
end

```

```

%% Find accumulative probability for each state, to be able to determine
%starting state
indeks = P(:, :, leg) > 0;
if state == 0 || indeks(state) == 0
    LongRunP1 = P(:, :, leg) ^ 1000;
    PStart = zeros(length(LongRunP1), 1);
    PStart(1) = LongRunP1(1, 1);
    for i = 2:length(LongRunP1)
        PStart(i) = PStart(i-1) + LongRunP1(i, i);
    end

    %% Determine starting state
    r = Random_Values(randnum, col);
    randnum = randnum + 1;
    count = 1;

    if r >= 1
        count = length(P(:, :, 1));
    else

        while r > PStart(count)
            count = count + 1;
        end
    end

    state = count;

else
    %% Find next state from previous

    cum_dist = cumsum(P(state, :, leg));

    iter = 0;
    r = Random_Values(randnum, col);
    randnum = randnum + 1;
    temp = find(cum_dist > r);

    while isempty(temp) == 1 && iter < 100
        r = Random_Values(randnum, col);

```

```
        randnum=randnum+1;
        temp = find(cum_dist>r);
        iter = iter + 1;
    end

    if isempty(temp) == 1
        temp = state;
    end
state = temp(1);

end

if three_mat==1
    Sea_HSS = state;
elseif three_mat==2
    Sea_HSDS = state;
else
    Sea_TPS = state;
end

end

end
```

B.4 Hollenbach

```
% Function that uses Hollenbachs method, and Kreitner to find resistance%
% Author: Peter Tenfjord and Martin Bakke
% Date: 01.06.2017
% Reference is made to (Steen and Minsaas, 2013)
```

```
function [R, Raw] = hollenb(Vreq, Hst, num)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%           Hollenbach.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%output: R [N] and Raw [N]
% Ship velocities in m/s
Vs = Vreq*0.5144;
```

```
%constants
rho = 1025;
gravk = 9.81; %Gravity
nu = 1.1395E-6; % viscosity
```

```
%ship particulars from vessel info
Los = num(1);
L = num(2);
Lwl = num(3);
B=num(4);
CB=num(8);
k=num(9);
```

```
T = num(6);
TA = num(6);
TF = num(7);
Dp = num(10);
NRud = num(11);
NBrac = num(12);
NBoss = num(13);
NThr = num(14);
S = num(15);
```

```

Lfn=Los;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Constants from Hollenbachs paper:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 'Mean' resistance coefficients
a = num(16:25);
%a1 means a(1) and so on
b = [num(26:28);          %b12 means b(1,2)
      num(29:31);
      num(32:34)];

d = num(36:38);
e = num(39:40);
f = num(41:43);
g = num(44:46);

% Froude's number
Fn = Vs/sqrt(gravk*Lfn);
Fnkrit = d*[1 CB CB^2]';

c1 = Fn/Fnkrit;
c1_min = 0;

Rns = Vs*L/nu;                                     % Reynold's number
- for ship
CFs = 0.075/(log10(Rns)-2)^2;                       % ITTC friction line for
- ship

% Calculation of C_R for given ship
% Mean value
CRFnkrit = max(1.0,(Fn/Fnkrit)^c1);

kL = e(1)*L^(e(2));

% There is an error in the hollenbach paper and in Minsaas' 2003 textbook, which
% is corrected in this formula by dividing by 10
CRstandard = [1 CB CB^2]*(b*[1 Fn Fn^2]')/10;

CR_hollenbach = CRstandard*CRFnkrit*kL*prod([T/B B/L Los/Lwl Lwl/L (1+(TA-TF)/L)
- ...

```



```

Dp/TA (1+NRud) (1+NBrac) (1+NBoss) (1+NThr)].^a);

CR = CR_hollenbach*B*T/S; % Resistance coefficient,
- scaled for wetted surface
C_Ts = CFs + CR; % Total resistance coeff.
- ship
R_T_mean = C_Ts*rho/2*Vs^2*S; % Total resistance to the
- ship

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% When accounting for form factor, roughness and correlation coef., as given by
- Minsaas
Rnm = 6*sqrt(6/L)*10^6/1.1395*Vs; % Reynold's number for model
CFm = 0.075/(log10(Rnm)-2)^2; % ITTC friction line for
- model
dCF = (110.31*(150*Vs/0.514)^0.21 - 403.33)*CFs^2; % Increase in friction due to
- roughness
CA = -0.228*10^(-3); % Correlation
- coefficient
CR_2 = CR_hollenbach*B*T/S - k*CFm; % Resistance coefficient
C_Ts_2 = (1+k)*(CFs + dCF) + CR_2 + CA; % Total resistance coeff. ship
R_T_mean_2 = C_Ts_2*rho/2*Vs^2*S; % Total resistance to the ship

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

R = max(R_T_mean_2,R_T_mean); %N

% estimate of the added resistance due to waves, from Kreitner (ITTC 2005)
% Should only be used on waves up to two meters in height.
% If waves are larger, then a Raw is taken as a conservative 20% of R

if Hst<=2
    Raw=(0.64*Hst^2*num(4)^2*num(8)*rho*gravk/num(3)); %N
else
    Raw=0.20*R;
end
end

```

B.5 Interaction with ShipX - Veres

The codes in this section are required to run Veres through batch mode, and interpret the results.

B.5.1 Calculating added resistance

```

%% Calculates total resistance. Uses waveres results for calm water resistance, done by
- interpolation.
% Calculates added resistance by running Veres in batch mode.
%
% Author:    Martin Bakke and Peter Tenfjord
% Date:     01.06.2017

function [R, Raw] = run_shipx(Vreq, Hs_sea, Hsd_sea, Tp_sea, Hs_swl, Hsd_swl, Tp_swl,
- Resistance_Table, num)
% R - calm water resistance
% Raw - added resistance
%% Parameters and Constants
rho = 1025; % density of sea water [kg/m^3]
g = 9.81; % acceleration [m/s^2]
Lpp = num(1); % Vessel length perpendiculars [m]
B = num(4); % Vessel beam [m]

%% For Simulink these are needed
coder.extrinsic('system')
coder.extrinsic('fopen')
coder.extrinsic('fgetl')
coder.extrinsic('str2num')
coder.extrinsic('delete')
coder.extrinsic('dlmread')
coder.extrinsic('num2str')

%% Making the direction between 0 and 180 degrees

Hsd_sea = acosd(cosd(Hsd_sea));
Hsd_swl = acosd(cosd(Hsd_swl));

%% Veres run for sea
InputWrite(Vreq,Hs_sea,Hsd_sea);

```

```

system('C:\Users\Martin\OneDrive - NTNU\Matlab\Proggng_master\RunVeres.bat'); %
    - Running Veres in batch mode
Sea_Vec = zeros(31,10);
Sea_Vec = dlmread('input.re2','','',10); % Reading output file
delete('input.hyd','input.log','input.out','input.re1','input.re2','input.str','input.sw-
    - y','input.veres')

% len = length(Sea_Vec);
len = 31;
freq_sea = zeros(len,1);
sup_sea = zeros(len,1);
S_sea = zeros(len,1);
CRAW_sea = zeros(len,1);
ADDEDR_sea = 0;
C_sea = zeros(len,1);

freq_sea = Sea_Vec(1:len,1); % Reading frequency vector
S_sea = wavespec(4,[Hs_sea, Tp_sea],freq_sea,0); % Creating wavespectrum, 4 is for
    - Pierson-Moskowitz

CRAW_sea = Sea_Vec(1:len,2); % Added resistance operator from shipx
sup_sea = times(S_sea,CRAW_sea); % Multiplying added resistance operator with
    - wavespectrum
ADDEDR_sea = 2*trapz(freq_sea,sup_sea); % Integrating over the frequency domain
if Hs_swl == 0
    Raw_sea = 0;
else
    Raw_sea = (ADDEDR_sea*rho*g*B^2)/Lpp; % Calculating added resistance
end

%% Veres run for swell
InputWrite(Vreq,Hs_swl,Hsd_swl);

system('C:\Users\Martin\OneDrive - NTNU\Matlab\Proggng_master\RunVeres.bat');
Swl_Vec = zeros(31,10);
Swl_Vec = dlmread('input.re2','','',10);
delete('input.hyd','input.log','input.out','input.re1','input.re2','input.str','input.sw-
    - y','input.veres')

len = length(Swl_Vec);
freq_swl = zeros(len,1);
sup_swl = zeros(len,1);

```

```
S_swl = zeros(len,1);
CRAW_swl = zeros(len,1);
ADDEDR_swl = 0;
C_swl = zeros(len,1);

freq_swl = Swl_Vec(1:len,1);
S_swl = wavespec(4,[Hs_swl, Tp_swl],freq_swl,0);

CRAW_swl = Swl_Vec(1:len,2);
sup_swl = times(S_swl,CRAW_swl);
ADDEDR_swl = 2*trapz(freq_swl,sup_swl);
if Hs_swl == 0
    Raw_swl = 0;
else
    Raw_swl = (ADDEDR_swl*rho*g*B^2)/Lpp;
end

%% Calculation of Resistance
R = interp1(Resistance_Table(:,1),Resistance_Table(:,2),Vreq); % Calm water resistance

Raw = Raw_sea + Raw_swl; % Added resistance

end
```

B.5.2 .bat file to initiate Veres through batch mode with the correct input

```
% First line represents the path to the batch file  
% Second line is to await licensed use  
% Third line specifies exe file and input file  
  
set PATH=%PATH%;C:\Program Files (x86)\ShipX\PlugIns\Veres\  
set ARGS=/b /belownormal /wait  
start %ARGS% veres_batch.exe 1 input.veres
```

B.6 Organization of vessel parameters

In table B.1, the organization of vessel parameters are showed. The Matlab codes get information about the vessel here, that is used for the calculations during the simulation.

Table B.1: Location of parameters in the excel spreadsheet: Vessel info

Location	Parameter	Unit
2	LOA	[m]
3	Lwl	[m]
4	B	[m]
5	T	[m]
6	TA	[m]
7	TF	[m]
8	C_B	[-]
9	k	[-]
10-14 & 16-47	Constants for Hollenbach	[-]
15	S	[m^2]
48-53	Engine specifics	[-]
54	∇	[m^3]
55	S_B	[m^2]

C. Data Sheets from ShipX

DATA-CHECK PROPERTIES	ENCL.	1)
	REPORT	
	DATE	2017-06-09
	REF	

Run name: Case

Ship name: contour.lwl92.dxf (imported)
 Loading condition description: Design waterline

ShipX exported data
 Main dimensions (from input):

Length between perpendiculars	(m)	200.000
Breadth	(m)	32.280
Draught, midship	(m)	8.833
Sinkage	(m)	0.000
Trim, + = aft	(deg)	-0.102

Coefficients for data check etc.:

Type		Specified	Calculated
Displacement	(tonnes)	43704.99	43448.54*
Vertical center of bouyancy,	KB		4.583*
Vertical center of gravity,	VCG	13.100*	
Longitudinal center of bouyancy,	LCB		105.522*
Longitudinal center of gravity,	LCG	105.523	105.522*
Block coefficient,	Cb	0.745	0.743
Water plane area coefficient,	Cw	0.835	0.813
Prismatic coefficient,	Cp		0.748
Mid section area coefficient,	Cm	1.002	0.993*
Longitudinal metacentric height,	GMl		280.518*
Transverse metacentric height,	GMt		0.999*
Roll radius of gyration,	r44	11.298*	
Pitch radius of gyration,	r55	50.000*	
Yaw radius of gyration,	r66	50.000*	
Roll-yaw radius of gyration,	r46	0.000*	

* - Applied in the hydrodynamic calculations

Figure C.1: Case condition

DATA-CHECK PROPERTIES	ENCL.	2)
	REPORT	
	DATE	2017-06-09
	REF	

Run name: Design Condition

Ship name: Design Cond
 Loading condition description: Design waterline

ShipX exported data
 Main dimensions (from input):

Length between perpendiculars	(m)	204.600
Breadth	(m)	32.280
Draught, midship	(m)	12.000
Sinkage	(m)	0.000
Trim, + = aft	(deg)	0.000

Coefficients for data check etc.:

Type		Specified	Calculated
Displacement	(tonnes)	61511.98	61165.10*
Vertical center of bouyancy,	KB		6.287*
Vertical center of gravity,	VCG	12.680*	
Longitudinal center of bouyancy,	LCB		102.679*
Longitudinal center of gravity,	LCG	102.679	102.679*
Block coefficient,	Cb	0.754	0.753
Water plane area coefficient,	Cw	0.879	0.859
Prismatic coefficient,	Cp		0.757
Mid section area coefficient,	Cm	0.997	0.995*
Longitudinal metacentric height,	GM1		256.184*
Transverse metacentric height,	GMt		1.013*
Roll radius of gyration,	r44	11.298*	
Pitch radius of gyration,	r55	51.150*	
Yaw radius of gyration,	r66	51.150*	
Roll-yaw radius of gyration,	r46	0.000*	

* - Applied in the hydrodynamic calculations

Figure C.2: Design condition

DATA-CHECK PROPERTIES	ENCL.	3)
	REPORT	
	DATE	2017-06-09
	REF	

Run name: Vessel 1

Ship name: Design Cond
 Loading condition description: TrimAft

ShipX exported data
 Main dimensions (from input):

Length between perpendiculars	(m)	204.600
Breadth	(m)	32.280
Draught, midship	(m)	12.000
Sinkage	(m)	0.000
Trim, + = aft	(deg)	-0.280

Coefficients for data check etc.:

Type		Specified	Calculated
Displacement	(tonnes)	64734.82	61428.87*
Vertical center of bouyancy,	KB		6.310*
Vertical center of gravity,	VCG	12.680*	
Longitudinal center of bouyancy,	LCB		101.386*
Longitudinal center of gravity,	LCG	100.972	101.385*
Block coefficient,	Cb	0.751	0.756
Water plane area coefficient,	Cw	0.872	0.867
Prismatic coefficient,	Cp		0.760
Mid section area coefficient,	Cm	1.011	0.995*
Longitudinal metacentric height,	GML		260.638*
Transverse metacentric height,	GMt		1.081*
Roll radius of gyration,	r44	11.298*	
Pitch radius of gyration,	r55	51.150*	
Yaw radius of gyration,	r66	51.150*	
Roll-yaw radius of gyration,	r46	0.000*	

* - Applied in the hydrodynamic calculations

Figure C.3: Vessel 1

DATA-CHECK PROPERTIES	ENCL.	4)
	REPORT	
	DATE	2017-06-09
	REF	

Run name: Vessel 2

Ship name: Design Cond
 Loading condition description: TrimFore

ShipX exported data
 Main dimensions (from input):

Length between perpendiculars	(m)	204.600
Breadth	(m)	32.280
Draught, midship	(m)	12.000
Sinkage	(m)	0.000
Trim, + = aft	(deg)	0.280

Coefficients for data check etc.:

Type		Specified	Calculated
Displacement	(tonnes)	61776.98	61429.01*
Vertical center of bouyancy,	KB		6.310*
Vertical center of gravity,	VCG	12.680*	
Longitudinal center of bouyancy,	LCB		101.385*
Longitudinal center of gravity,	LCG	101.386	101.386*
Block coefficient,	Cb	0.757	0.756
Water plane area coefficient,	Cw	0.884	0.867
Prismatic coefficient,	Cp		0.760
Mid section area coefficient,	Cm	1.004	0.995*
Longitudinal metacentric height,	GMl		260.640*
Transverse metacentric height,	GMt		1.081*
Roll radius of gyration,	r44	11.298*	
Pitch radius of gyration,	r55	51.150*	
Yaw radius of gyration,	r66	51.150*	
Roll-yaw radius of gyration,	r46	0.000*	

* - Applied in the hydrodynamic calculations

Figure C.4: Vessel 2

DATA-CHECK PROPERTIES	ENCL.	5)
	REPORT	
	DATE	2017-06-09
	REF	

Run name: Vessel 3

Ship name: Alt.1
 Loading condition description: Design waterline

ShipX exported data
 Main dimensions (from input):

Length between perpendiculars	(m)	204.600
Breadth	(m)	33.280
Draught, midship	(m)	12.000
Sinkage	(m)	0.000
Trim, + = aft	(deg)	0.000

Coefficients for data check etc.:

Type		Specified	Calculated
Displacement	(tonnes)	61511.98	63056.55*
Vertical center of bouyancy,	KB		6.287*
Vertical center of gravity,	VCG	13.100*	
Longitudinal center of bouyancy,	LCB		102.679*
Longitudinal center of gravity,	LCG	102.679	102.679*
Block coefficient,	Cb	0.754	0.753
Water plane area coefficient,	Cw	0.879	0.859
Prismatic coefficient,	Cp		0.757
Mid section area coefficient,	Cm	0.997	0.995*
Longitudinal metacentric height,	GMl		255.776*
Transverse metacentric height,	GMt		1.060*
Roll radius of gyration,	r44	11.298*	
Pitch radius of gyration,	r55	51.150*	

Figure C.5: Vessel 3