



Norwegian University of
Science and Technology

Digital Twin of Vessels in Arctic Environments

Extending a Simulation Environment to allow
for External Control of Multiple Vessels

Preben Jensen Hoel
Andre Nilsson Rolandsen

Master of Science in Engineering and ICT

Submission date: June 2018

Supervisor: Dong Trong Nguyen, IMT

Co-supervisor: Jon Bjørnø, IMT

Norwegian University of Science and Technology
Department of Marine Technology



MSC THESIS DESCRIPTION SHEET

Name of the candidate:	Preben Jensen Hoel André Nilsson Rolandsen
Field of study:	Marine control engineering
Thesis title (Norwegian):	Digital Tvilling av Fartøyer i Arktiske Farvann – Utvidelse av et Simuleringsmiljø for å muliggjøre Ekstern Regulering av Fartøyer
Thesis title (English):	Digital Twin of Vessels in Arctic Environments – Extending a Simulation Environment to allow for External Control of Multiple Vessels

Background

The oil and gas exploration and exploitation have been moving to more remote areas, and is now moving more and further into Arctic waters. The reasons being potentially large oil and gas reserves and warmer climate exposing more areas that were inaccessible before, making these areas available for petroleum extraction. One of the major challenges to the development in these waters are that the water depths often exceed depths where bottom-founded structures become impractical, hence stationary floating vessels are required. The challenge then becomes how to keep a vessel in place using DP or how to maneuver the vessel in managed sea ice.

In addition to the oil and gas industry the shipping and cruise industry have been showing an increased interest for marine vessels operating in Arctic environments/sea ice. One of the main motivations for the shipping industry is the use of the shorter Northeast passage between Europe and Asia which reduces the travel time. For the cruise industry the major motivations have been more accessible areas and a general increase in the tourist industry over the last few years.

The design of a proper control system for a vessel operating in an ice-covered sea is highly dependent on an accurate model of the ice loads. Each scenario of ice-vessel interactions, i.e. slushed ice, broken ice, level ice, ridge ice and icebergs, needs to be studied on an ad-hoc basis. Among these, managed ice interaction with vessel is considered as the most frequent scenarios for a vessel's operations in the Arctic region.

Work description

1. Perform a background and literature review to provide information and relevant references on:
 - Loads from sea ice, and sea ice in drift.
 - Simulation of vessels in ice.
 - DP-operations in ice infested waters.
 - Digital Twin.
 - Control algorithms for DP vessels.
2. Write a list with abbreviations and definitions of terms, explaining relevant concepts related to the literature study and project assignment.
3. Implement a communication interface between C++ (SAMS) and Simulink (Controller):
 - The connection should be fast and reliable.
 - Should not put restrictions on Simulink.
 - Input to Simulink should be position, velocity and ice-forces.
 - Output from Simulink should be the controller forces.
4. Implement a communication interface between C++ (SAMS) and an external software for manual control of the vessel:
 - The input and output should be general and changeable by the programmer.
 - The communication should take place over TCP.
5. Implement the possibility to use several vessels simultaneously:
 - The vessels will operate independently of each other.
6. Improvement of SAMS in terms of:
 - Hydrodynamic forces:
 - Should be calculated in Simulink.
 - Removal of ice floes:
 - Remove ice floes when they are out of the domain, or below a certain mass.
 - Improve simulator performance.
 - Log the removed floes.
7. Implement, test and compare different control strategies in Simulink:
 - The control strategies should include:
 - Simple controller.
 - Observer.
 - Reference model.
 - Simple thrust allocation.
8. Test how the controller handles disturbances and/or loss of input signal.
9. Test and verify the simulator against empirical data:
 - Ice resistance models.
 - OATRC 2015.
10. Implement and test an IM control strategy:
 - One vessel will remain in DP.
 - The other will follow a predefined IM path.

Specifications

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

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

The report shall be organized in a logical structure to give a clear exposition of background, results, assessments, and conclusions. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 10 pts., and it is not expected to be longer than 120-160 A4 pages, from introduction to conclusion, unless otherwise agreed upon. It shall be written in English (preferably US) and contain the following elements: Title page, abstract, acknowledgements, thesis specification, list of symbols and acronyms, table of contents, introduction with objective, background, and scope and delimitations, main body with problem formulations, derivations/developments and results, conclusions with recommendations for further work, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to the main supervisor, with the printed copy signed by the candidate, and otherwise according to NTNU procedures. The final revised version of this thesis description must be included. Computer code, pictures, videos, data series, and a PDF version of the report shall be enclosed electronically with all submitted versions.

Start date: January 15th, 2018 **Due date:** June 18th, 2018

Supervisor: Dong Trong Nguyen
Co-advisor(s): Jon Bjørnø

Trondheim, June 7th 2018



Dong Trong Nguyen
Supervisor

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This master thesis is a continuation of the project thesis delivered in December 2017. The master thesis has been conducted as part of the study program Engineering and ICT at NTNU in Trondheim during the spring of 2018.

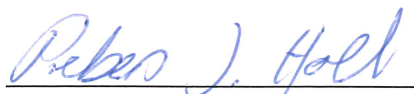
The thesis have been both challenging and exciting and spans multiple fields of studies, including computer science, cybernetics, marine technology and informatics.

It have caused some frustrations as the source code of SAMS was first made available late in the process, additionally some updates to the source code was challenging to implement without extensive modifications. Working with a developer edition of the source code also made it necessary to test and verify the simulator itself, and not just our modifications, which at times was both frustrating and seemed never-ending.

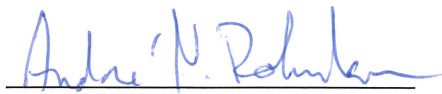
However the thesis also gave an immense sensation of accomplishment as things started to work, and gave a good insight into working life. To be able to work with a state-of-the-art simulator and fix problems as they have arisen also gave a good learning outcome.

After the work on this thesis was concluded, a large number of the modifications implemented in this thesis have been implemented into the latest SAMS version, scheduled for release in the end of June 2018. The main task of the thesis, the TCP interface, is scheduled to be released with a later version of SAMS, but is already in use for the doctoral thesis of Jon Bjørnø.

The readers should have good knowledge on control theory as well as basic knowledge of vessel dynamics, programming and network communication, knowledge on ice loads would be beneficial.



Preben Jensen Hoel



André Nilsson Rolandsen

Trondheim, June 11, 2018

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgment

We would especially like to thank our supervisor Dong Trong Nguyen for his help and support during the thesis. We would also like to thank him for the large contributions and many discussion regarding the motion control system and for helping with the structure and content of this thesis.

Secondly we would like to give a special thanks to our co-advisor Jon Bjørnø for the long discussions, ideas and feedback regarding both the motion control systems, the simulator and additional features. We would also like to thank Jon for his contribution in making the vessel model used in the simulator and for the dynamic model of the vessel. In addition a special thanks for constructive feedback on the thesis and its structure.

We would also like to thank ArcISO with Raed Lubbad and Sveinung Løset together with their employees for their support and feedback during the spring and for giving us access to the source code of SAMS. In addition we would express a special thanks for helping to guide us toward a professional code and for answering questions regarding both the code and the methods behind the simulator.

We would also like to thank Marnix van den Berg, Andrei Tsarau and Wenjun Lu for help with the SAMS code and for discussions regarding the implementations. We would also like to thank Wenjun Lu for sharing his code for making ice fields, and give a special recognition to Marnix van den Berg for helping solve problems regarding our implementations and modifications in SAMS.

At last a thanks to Sören Ehlers and Sandro Erceg for help with ice loads and the empirical level ice models.

Lastly we recognize Exxon mobile and SAMCoT for their contribution with full-scale data for the OATRC 2015 expedition, used to validate the simulator in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

Abstract

This thesis presents the work performed on extending and improving a numerical ice tank simulator developed by ArcISO AS called Simulator of Arctic Marine Structures (SAMS). The main work for this thesis has been on extending SAMS, so external software can connect to, and control, a vessel in SAMS. To allow for such a connection, SAMS has been modified to work as a server which clients in external software can connect to over a TCP/IP connection. The simulator itself was also modified with the added ability for multiple vessels as well as several minor improvements. An external software for manual control of a vessel was also created.

The main external software used in this thesis have been Simulink, and several controllers have been implemented in Simulink for the various simulations performed in this thesis. These controllers includes a PID-, an acceleration feedback-, a velocity- and an ice management-controller. With these controllers a nonlinear observer and a thruster allocation was also implemented. Simulations regarding ice management and dynamic positioning have been conducted together with simulations to verify the simulator against full-scale sea trial data and empirical formulas.

THIS PAGE INTENTIONALLY LEFT BLANK

Sammendrag

Denne avhandlingen presenterer arbeidet med å utvide og forbedre en numerisk istank-simulator utviklet av ArcISO AS, kalt Simulator of Arctic Marine Structures (SAMS). Hovedarbeidet har vært å utvide SAMS slik at ekstern programvare kan koble til og regulere et fartøy i SAMS. For å tillate en slik tilkobling har SAMS blitt modifisert til å fungere som en server, der klienter i ekstern programvare kan koble til over en TCP/IP tilkobling. Simulatoren har i tillegg blitt modifisert med mulighet for flere fartøy, samt flere mindre forbedringer. En ekstern programvare for manuell regulering av et fartøy ble også laget.

Den viktigste eksterne programvaren som brukes i denne oppgaven, har vært Simulink, og flere forskjellige regulatorer har blitt implementert i Simulink for de ulike simuleringene som utføres i denne avhandlingen. Disse regulatoren inkluderer en PID-, en akselerasjons tilbakeført-, en hastighets- og en ishåndterings-regulator. Med disse regulatoren ble en ulineær observator og en thruster allokering også implementert. Simuleringer vedrørende ishåndtering og dynamisk posisjonering er utført sammen med simuleringer for å verifisere simulatoren mot fullskala data og empiriske formler.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

MSc Thesis Description	i
Preface	v
Acknowledgment	vii
Abstract	ix
Sammendrag	xi
Abbreviations	xxv
Terminology	xxxii
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Literature Review	2
1.4 Objective	3
1.5 Scope	3
1.6 Organization of Project	5
2 Literature Review	7
2.1 Ice Loads from Level Ice	7
2.1.1 Ice Resistance	8
2.1.2 Simulation of Ice Loads	10
2.1.3 Ice Drift	10
2.2 Control Systems	11

2.2.1	Ice Navigation	12
2.3	Historic DP Operations in Ice Infested Waters	13
2.4	Simulation of Vessels in Ice Infested Waters	14
2.5	Digital Twins	15
3	Ice Forces and Vessel Dynamics	17
3.1	Ice Resistance Models	17
3.1.1	Level ice	17
3.1.1.1	Vance (1980)	18
3.1.1.2	Lindqvist (1989)	18
3.1.1.3	Keinonen and Browne (1991); Keinonen et al. (1996)	18
3.1.1.4	Riska (1998)	19
3.1.2	Managed Ice Loads	19
3.2	Ice Force Estimation	20
3.3	Model Kinematics	20
3.4	Model Kinetics	23
3.5	Hydrostatics	26
3.6	Hydrodynamics	28
3.6.1	Potential Theory	28
3.6.2	Seakeeping Theory	29
3.6.3	Maneuvering Theory	29
3.6.3.1	Added Mass and Coriolis	30
3.6.3.2	Damping	30
3.7	Summary of Vessel Dynamics	31
4	Motion Control Systems	33
4.1	Guidance Navigation and Control	33
4.1.1	Computer-Based Control Systems	34
4.2	Control System	36
4.2.1	Acceleration Feedback	37
4.2.2	Tuning of the Controller	37
4.3	Thruster Allocation	38
4.4	Filter	39
4.5	Reference Model	40
4.6	Observer	41
4.6.1	Nonlinear Passive Observer	41
5	Improvements of Simulator of Arctic Marine Structures	45

5.1	Simulator of Arctic Marine Structures	45
5.2	Modification of SAMS	47
5.2.1	Additional Vessels	47
5.2.2	Hydrodynamics	47
5.2.3	Removal of Floes	48
5.2.4	Simulator Output	48
5.2.5	Testing and Verification	49
5.3	Ice Fields	49
5.3.1	Making of Ice Field Sections	49
5.3.2	Stitching of Ice Fields	50
5.4	Note on Copyright of SAMS	50
6	Implementation of a Transmission Control Protocol Interface	51
6.1	TCP/IP	51
6.1.1	Socket	53
6.2	TCP Interface	54
6.2.1	Simulation Loop	54
6.2.2	Multiple Vessels	55
6.2.3	Serialization Format	55
6.2.3.1	RapidJSON	56
6.2.4	Failure Handling	57
7	Implementation and Verification of Motion Control Systems	59
7.1	Simulink Model	59
7.2	Reference Model	60
7.2.1	Setpoint Reference Model	61
7.2.2	Velocity Reference Model	61
7.2.2.1	Verification of the Velocity Reference Model	62
7.2.3	Ice Management Reference Model	62
7.3	Observer	64
7.3.1	Verification of Observer	64
7.4	Controller	66
7.4.1	Verification of the Controller	67
7.4.2	Acceleration Feedback Controller	68
7.4.3	Sway Compensation	68
7.4.4	Ice Management Controller	68
7.5	Thruster Allocation	69
7.5.1	Thruster Dynamics	69
7.6	Hydrodynamics	70

7.6.1	Tuning of the Model	70
7.6.2	Verification of Hydrodynamics	71
7.7	Manual Control	72
7.7.1	Website	72
7.7.2	JavaScript Client	74
7.7.3	Proxy Server	74
8	Simulation Setup	75
8.1	General Setup	75
8.1.1	Vessel	76
8.2	Discussion of the Modifications and Assumptions of Oden	76
9	Validation of SAMS in Terms of Level Ice Resistance	79
9.1	Setup	79
9.1.1	Ice Field	80
9.2	Results	80
9.3	Discussion	80
10	OATRC 2015 Transit Simulation	83
10.1	Setup	83
10.1.1	Ice Field	84
10.2	Results	84
10.3	Discussion	85
10.3.1	Validation Against Empirical Data	85
10.3.2	Controller Performance	87
10.3.3	Validation Against Palmer and Croasdale (2014)	87
10.3.4	Validation Against Kjerstad et al. (2018)	88
11	Transit Acceleration Feedback Simulation	89
11.1	Setup	89
11.2	Results	89
11.3	Discussion	90
12	Dynamic Positioning and Ice Management Simulation	93
12.1	Setup	93
12.1.1	Ice Field	94
12.2	Dynamic Positioning without Ice Management	94
12.2.1	Results	94
12.2.2	Discussion	95
12.3	Dynamic Positioning with Ice Management	97

12.3.1	Results	97
12.3.1.1	Dynamic Positioning Vessel	98
12.3.1.2	Ice Management Vessel	98
12.3.2	Discussion	99
12.3.2.1	Dynamic Positioning Vessel	100
12.3.2.2	Ice Management Vessel	101
13	Discussion of Simulator Improvements and Modifications	105
13.1	SAMS and the TCP Interface	105
13.1.1	Performance Enhancements	106
13.1.2	Vessel Interaction	107
13.1.3	Hydrodynamics	107
13.1.4	Possibilities with TCP Interface	108
13.2	Manual Control	108
14	Conclusion	111
14.1	Further Work	112
	Bibliography	115
A	Ice Fields	I
B	Ice Resistance Models	III
B.1	Lindqvist (1989)	III
B.2	Keinonen and Browne (1991); Keinonen et al. (1996)	IV
B.3	Riska (1998)	VI
C	Oden Data	VII
D	Results	IX
D.1	Observer Verification	IX
D.2	OATRC 2015 Transit Simulation	X
D.3	Transit Acceleration Feedback Simulation	XI
D.4	Dynamic Positioning without Ice Management	XII
D.5	Dynamic Positioning with Ice Management	XIV
D.5.1	Dynamic Positioning Vessel	XIV
D.5.2	Ice Management Vessel	XVI
E	Documentation	XIX
E.1	<i>RHAS.itconfig</i>	XIX

E.2	Simulink	XXII
E.3	Web Application Documentation	XXIV
E.4	Output	XXVI
F	Simulink Model	XXIX
F.1	Color Coding	XXIX
F.2	Main Model	XXX
F.3	Reference Models	XXXI
F.4	Observer	XXXIV
F.5	Controllers	XXXV
F.6	Thrust Allocation	XXXVIII
F.7	Hydrodynamics	XXXIX
G	Code	XLI
G.1	<i>SAMS.itconfig</i>	XLI
G.2	<i>RHAS.itconfig</i>	XLIII
G.3	Ice Stitcher	XLIV
H	Contents of Digital Appendix	XLVII

List of Tables

3.1	SNAME notation.	21
7.1	Saturation values for Oden in SAMS.	60
7.2	Natural frequency and relative damping for the reference model.	61
7.3	Gains and constants for the observer.	66
7.4	PID tuning values.	68
7.5	Parameters used in hydrodynamic models.	71
7.6	Time constants for LP filter in proxy server.	74
9.1	Ice parameters used in calculations by the level ice resistance models.	80
10.1	Coefficients used in Croasdale resistance models.	84
12.1	Data used to create the DP and IM ice field.	95
B.1	Table of parameters required for the Lindqvist resistance model.	IV
B.2	Table of parameters required for the Keinonen resistance model.	V
B.3	Constants in Riska resistance model, courtesy of Hu and Zhou (2016).	VI
B.4	Table of parameters required for the Riska resistance model.	VI
C.1	Vessel parameters for Oden.	VII
E.1	Output format provided by the simulator.	XXVI
E.2	Output format for the removed floes output file.	XXVII
F.1	Color coding scheme for Simulink blocks.	XXIX

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

2.1	Illustration of the ice-hull interaction process. Courtesy of Riska (2010b).	8
2.2	Illustration of time history of ice forces contributing to ice resistance defined as an average longitudinal load. Courtesy of Riska (2010a).	9
4.1	Illustration of a simplified GNC signal flow.	34
4.2	Illustration of a general marine control structure. Courtesy of Sørensen (2012).	35
4.3	Illustration showing digital control of an analog system. Courtesy of Sørensen (2012).	36
5.1	High level overview of the simulation environment together with the external software.	46
6.1	The Open Systems Interconnection model.	52
6.2	Detailed workflow of TCP interface from Figure 5.1.	55
7.1	Position output from velocity reference model and actual position of Oden.	62
7.2	Velocity output from velocity reference model and actual velocity of Oden.	62
7.3	Arched racetrack path used in the IM reference model.	63
7.4	Actual and estimated northing positions from the observer with white noise.	65
7.5	Actual and estimated surge velocities from the observer with white noise.	65

7.6	Actual and estimated northing positions from the observer during loss of signal simulation.	65
7.7	Actual and estimated surge velocities from the observer during loss of signal simulation.	65
7.8	Tracking performance of the controller compared to the setpoint reference model.	67
7.9	Comparison of hydrodynamic resistance for Oden.	71
7.10	Web application signal flow.	72
7.11	Website interface for manual control of Oden.	73
9.1	Predicted and simulated ice resistance in level ice.	81
10.1	Image 110 s into the OATRC 2015 transit simulation.	85
10.2	Predicted and simulated ice force in surge from OATRC simulation.	86
11.1	Image 360 s into the acceleration FB simulation with the reference controller.	90
11.2	Surge velocities from acceleration FB simulations.	91
11.3	Easting position from acceleration FB simulations.	91
12.1	Image 1800 s into the DP simulation without IM support.	95
12.2	Northing position of Oden during DP simulation without IM support.	96
12.3	Image 8310 s into the DP and IM simulation, here the lower vessel is the DP vessel and the upper vessel is the IM vessel.	97
12.4	Northing position of Oden during DP simulation with IM support.	98
12.5	Path of the IM vessel during DP simulation with IM support.	99
12.6	Surge velocity of the IM vessel during DP simulation with IM support.	99
A.1	Ice field used for OATRC 2015 transit and acceleration FB simulations.	I
A.2	Ice field used for DP and IM simulations.	II
C.1	Blueprint showing the modified thruster position of Oden.	VIII
D.1	Scaled version of Figure 7.4.	IX
D.2	Scaled version of Figure 7.5.	IX
D.3	Actual, estimated and unfiltered northing positions with 20 times normal white noise.	X
D.4	Actual and estimated velocities with 20 times normal white noise.	X

D.5	Actual and simulated surge velocities from OATRC simulation. . .	X
D.6	Predicted and simulated thruster forces in surge from OATRC simulation.	X
D.7	Filtered thruster forces in surge from acceleration FB simulations.	XI
D.8	Unfiltered thruster forces in surge in a smaller time interval from Figure D.7.	XI
D.9	Easting position of Oden during DP simulation without IM support.	XII
D.10	Heading of Oden during DP simulation without IM support. . . .	XII
D.11	Filtered ice force in surge of Oden during DP simulation without IM support.	XII
D.12	Filtered thruster force in surge of Oden during DP simulation without IM support.	XIII
D.13	Filtered ice force in sway of Oden during DP simulation without IM support.	XIII
D.14	Filtered thruster force in sway of Oden during DP simulation without IM support.	XIII
D.15	Easting position of Oden during DP simulation with IM support. .	XIV
D.16	Heading of Oden during DP simulation with IM support.	XIV
D.17	Filtered ice force in surge of Oden during DP simulation with IM support.	XIV
D.18	Filtered thruster force in surge of Oden during DP simulation with IM support.	XV
D.19	Filtered ice force in sway of Oden during DP simulation with IM support.	XV
D.20	Filtered thruster force in sway of Oden during DP simulation with IM support.	XV
D.21	Sway velocity of IM vessel during DP simulation with IM support.	XVI
D.22	Crab angle of IM vessel during DP simulation with IM support. .	XVI
D.23	Filtered ice force in surge of IM vessel during DP simulation with IM support.	XVI
D.24	Filtered thruster force in surge of IM vessel during DP simulation with IM support.	XVII
D.25	Filtered ice force in sway of IM vessel during DP simulation with IM support.	XVII
E.1	Overview Simulink template.	XXII
E.2	Inside the "send" block from Figure E.1.	XXIII
E.3	Web application signal flow.	XXIV
E.4	Illustration of setup of proxy server.	XXV

F.1	Overview of main model.	XXX
F.2	Overview of reference model.	XXXI
F.3	Overview of velocity reference model.	XXXII
F.4	Overview of IM reference model.	XXXIII
F.5	Overview of observer model.	XXXIV
F.6	Overview of DP PID controller.	XXXV
F.7	Overview of acceleration FB controller.	XXXVI
F.8	Overview of IM controller.	XXXVII
F.9	Overview of thruster allocation model.	XXXVIII
F.10	Overview of the hydrodynamics model from MSS toolbox.	XXXIX

Abbreviations

Abbreviations

ACEX Arctic Coring Expedition

A/D Analog to Digital

API Application Programming Interface

ArcISo Arctic Integrated Solutions AS

CB Center of Buoyancy

CFD Computational Fluid Dynamic

CG Center of Gravity

CO Origin of {b}

CPP Controllable Pitch Propeller

D/A Digital to Analog

DGPS Differential Global Positioning Systems

DNV GL Det Norske Veritas

DOF Degree of Freedom

DP Dynamic Positioning

EFEM Explicit Finite Element Method

ELBE Efficient Lattice Boltzman Enviroment

FB Feedback

FF Feedforward

GES Global Exponential Stability

GNC Guidance Navigation and Control

GNSS Global Navigation Satellite System

GPS Global Positioning System

GPU Graphics Processing Unit

GUI Graphical User Interface

H- ∞ H-infinity

HIL Hardware In The Loop

HSVA Hamburg Ship Model Basin

IM Ice Management

IMU Inertial Measurement Unit

IP Internet Protocol

JSON JavaScript Object Notation

LF Low Frequency

LOS Line of Sight

LP Low Pass

LQG Linear Quadratic Gaussian

LS Least-Squares

MIMO Multiple Input Multiple Output

NED North-East-Down

NIT Numerical Ice Tank

NMPC Nonlinear Model Predictive Controller

NRCan Natural Resources Canada

NTNU Norwegian University of Science and Technology

OATRC Oden Arctic Technology Research Cruise

OSI Open Systems Interconnection

PERD Program of Energy Research and Development

RBF Radial Basis Function

RPM Revolutions per minute

SAMCoT Sustainable Arctic Marine and Coastal Technology

SAMS Simulator of Arctic Marine Structures

SISO Single Input Single Output

TCP Transmission Control Protocol
TCP/IP Transmission Control Protocol/Internet Protocol
UDP User Datagram Protocol
UGES Uniform Global Exponential Stability
UKF Unscented Kalman Filter
UUV Unmanned Underwater Vehicle
WF Wave Frequency

Nomenclature

\dot{x} Time derivative operator
 x^{-1} Inverse operator
 x^\top Transpose operator
 \hat{x} Estimate operator
 \tilde{x} Estimated error operator ($x - \hat{x}$)
 ∇ Vector differential operator
 $\det()$ Determinant operator
 $A_{\omega p}$ Water plane area
 B Beam of vessel
 C_A Skew-symmetric Coriolis and centripetal matrix of the added mass
 C_{RB} Skew-symmetric Coriolis and centripetal matrix of the rigid body
 D Damping matrix
 δ Denoting tiny values close to zero
 Δ Denoting relative change in parameters
 E Young's modulus
 η Position and orientation vector
 η_d Desired position and orientation vector
 Θ_{nb} Euler angle between NED and BODY

f Linear force vector

g Gravity of Earth

G Generalised restoring matrix

\overline{GM}_i Metacentric height

h Ice thickness

$\mathbf{I}_{n \times n}$ $n \times n$ Identity matrix

$\mathbf{0}_{n \times m}$ $n \times m$ Zero matrix

L Length of vessel

m Moment vector

m Mass of object

M System inertia matrix

M_A Added mass matrix

M_i Metacenter

μ Fluid memory effects

μ_i Ice-ice friction

μ_{si} Structure-ice friction

ν Velocity vector

ν_c Water current velocity

ν_r Relative velocity between the vessel and the water

ω Angular velocity vector

ω_b Bandwidth frequency

ω_n Natural frequency

p Position vector

p_a Ambient ice pressure

p_c Ice field cohesion

ν Poisson's coefficient

R Rotation matrix from NED to BODY

\mathbf{R}^T Rotation matrix from BODY to NED

R_b Bending ice resistance

R_c Crushing ice resistance

R_{ice} Ice resistance

R_{ow} Open water resistance

R_s Submersion ice resistance

R_{total} Total resistance including both open water and ice resistance

ρ Volumetric mass density

s Laplace operator

\mathbf{S} Skew-symmetric matrix

σ_f Flexural strength of ice

T Vessel draft

$\boldsymbol{\tau}$ Force vector

\mathbf{u} Vector of control inputs

\mathbf{v} Linear velocity vector

V Speed of a vessel

ζ_i Relative damping ratio for DOF no. i

THIS PAGE INTENTIONALLY LEFT BLANK

Terminology

ACK Part of the three-way handshake in TCP, a small packet sent back from the recipient after receiving a TCP package confirming that the full package was correctly received.

Dynamic Positioning An operation to automatically maintain a vessel's position and heading by means of the vessel's own propellers and thrusters.

Ice Management An operation that involves several aspects which include observation and managing the ice field to reduce the possible maximum ice forces. The managing can be performed in several manners, e.g. crushing of ice floes and creating channels in the ice field, or towing huge ice floes/icebergs away from a critical area.

Internet Protocol The principal protocol on the Internet that provides the ability to send data across the network.

Numerical Ice Tank A numerical simulator made to simulate a towing carriage moving a structure through an ice field basin.

Serialization Format The process of translating data structures or object states into a format that can be stored or transmitted over a network.

Simulation Environment The simulator itself together with any interconnected parts required for the simulation to work on a fundamental level. In this thesis this will include SAMS and the TCP interface.

Socket Adress Refers to the tuple of an unique IP-address for a computer and a specific port number on that computer, put together it creates a unique address for connections.

Transmission Control Protocol A standard protocol which defines how to establish and maintain a connection, and allows for sending and receiving data over a network in a reliable manner without loss of information.

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

1.1 Background

The oil and gas exploration and exploitation have been moving to more remote areas, and is now moving more and further into Arctic waters. The reasons being potentially large oil and gas reserves and warmer climate exposing more areas that were inaccessible before, making these areas available for petroleum extraction. One of the major challenges to the development in these waters are that the water depths often exceed depths where bottom-founded structures become impractical, hence stationary floating vessels are required. The challenge then becomes how to keep a vessel in place using Dynamic Positioning (DP) or how to maneuver the vessel in managed sea ice.

In addition to the oil and gas industry the shipping and cruise industry have been showing an increased interest for marine vessels operating in Arctic environments/sea ice. One of the main motivations for the shipping industry is the use of the shorter Northeast passage between Europe and Asia which reduces the travel time. For the cruise industry the major motivations have been more accessible areas and a general increase in the tourist industry over the last couple of years.

The design of a proper control system for a vessel operating in an ice covered sea is highly dependent on an accurate model of the ice loads. Each scenario of ice-vessel interactions, i.e. slushed ice, broken ice, level ice, ridge ice and icebergs, needs to be studied on an ad-hoc basis. Among these, managed ice interaction is considered as the most frequent scenario for a vessel's operation in the Arctic region.

1.2 Motivation

Full size Arctic operations are expensive and very few are willing to perform research operations in ice fields, due to potential massive consequences both in the economical sense and for the environment. Model basin testing is an alternative to full-scale operations, however model testing in ice can be hard due to the limited locations where such tests are possible. These tests will not be 100 % accurate either since the model ice does not behave like natural ice, neither can perennial ice be simulated in these basins. Tests in these basins are also incredibly time consuming since only one or two test runs can be done in an ice field before the ice needs reconstruction. Other problems are that scaling factors and boundary effects will affect the model test. Despite these problems model testing is currently considered to be the state-of-the-art method for estimating global ice loads, Metrikin (2014).

Computer simulations are an alternative and provides the ability to run multiple simulations at a given time. With computer simulations there is the ability to get local ice loads acting on the vessel hull, they also open up the possibility to test different types of ice and maneuvers. Simulations have proven to be both faster and more cost effective than model testing, but their credibility are still low. This is mainly due to low amount of vessels being designed solely with the use of numerical simulators.

DP in open waters are quite common and extensively tested, however the loads on a DP vessel in ice is dramatically different than in open waters. One problem is the unevenly build up and rapid loss of loads depending on the ice field. There exists no good combination of a characteristic load model and sensor solutions to measure and estimate ice load on the vessel, Skjetne et al. (2014), hence a good simulator is essential to test and validate the robustness of a controller for DP vessel in ice.

1.3 Literature Review

A more comprehensive literature review can be found in Chapter 2.

Only a few DP operations have been performed in ice infested waters, among these are the coring operation in Northeast Greenland in 2008, Rohlén (2009), and the Sakhalin diving operation in 1999, Keinonen et al. (1999). However the operations have often had a large downtime and relied on heavy Ice Management (IM) support or mooring systems to stay within the designated area.

The simulation of vessels in ice infested waters is relative new and one of the first Numerical Ice Tank (NIT) simulators where proposed by Valanto and Puntigliano (1997).

Derradji-Aouat (2010) proposed a fully coupled NIT simulator and from this point many papers have been written on the subject. Among these are Metrikin (2014) who described a simulator for station keeping in sea ice and Lubbad and Løset (2011) who described a method for real-time simulation of vessel-ice interactions in level ice. This simulator was later developed into a NIT simulator where Denk (2016) made a DP control system in Python and connected it to the simulator using external communication protocols. However this solution was customized for one NIT simulator, hence it is difficult to generalize the solution and implement it into other simulators.

As a result of this, this thesis will focus on the development of a generalized communication interface between a control system and the Simulator of Arctic Marine Structures (SAMS) simulator, a NIT simulator, and making the communication interface in such a way that it can easily be implemented into other simulators. The work can be seen as a similar approach to the work carried out by Denk (2016), however both the simulator and communication protocols have been changed. This thesis also goes further compared to Denk (2016) where the control system should be able to not only simulate DP operations, but also maneuvering, and should support multiple external software and be able to work with multiple vessels simultaneously.

1.4 Objective

The objective of this master thesis is to extend a NIT simulator by implementing an interface which external software can connect to and control the movement of a vessel. A second objective is to test different controllers in ice conditions and compare their performance.

1.5 Scope

1. Perform a background and literature review to provide information and relevant references on:
 - Loads from sea ice, and sea ice in drift.
 - Simulation of vessels in ice.
 - DP operations in ice infested waters.
 - Digital Twin.
 - Control algorithms for DP vessels.

2. Write a list with abbreviations and definitions of terms, explaining relevant concepts related to the literature study and project assignment.
3. Implement a communication interface between C++ (SAMS) and Simulink (Controller):
 - The connection should be fast and reliable.
 - Should not put restrictions on Simulink.
 - Input to Simulink should be position, velocity and ice forces.
 - Output from Simulink should be the controller forces.
4. Implement a communication interface between C++ (SAMS) and an external software for manual control of the vessel:
 - The input and output should be general and changeable by the programmer.
 - The communication should take place over TCP.
5. Implement the possibility to use several vessels simultaneously:
 - The vessels will operate independently of each other.
6. Improvement of SAMS in terms of:
 - Hydrodynamic forces:
 - Should be calculated in Simulink.
 - Removal of ice floes:
 - Remove ice floes when they are out of the domain, or below a certain mass.
 - Improve simulator performance.
 - Log the removed floes.
7. Implement, test and compare different control strategies in Simulink:
 - The control strategies should include:
 - Simple controller.
 - Observer.
 - Reference model.
 - Simple thrust allocation.

8. Test how the control system handles disturbances and/or loss of input signal.
9. Test and verify the simulator against empirical data:
 - Ice resistance models.
 - OATRC 2015.
10. Implement and test an IM control strategy:
 - One vessel will remain in DP.
 - The other will follow a predefined IM path.

1.6 Organization of Project

Chapter 2 presents a comprehensive literature review concerning the most critical aspects about this thesis. This includes ice loads and how they have been simulated, control systems and their use in Arctic environments and lastly a review of the concept digital twins.

Chapter 3 presents theory behind the estimation of ice forces and some different ice resistance prediction models. The last part of the chapter will present the vessel dynamics for a vessel in motion, which includes kinematics, kinetics, hydrostatics and hydrodynamics.

Chapter 4 will go through the theory of a motion control system. This includes a controller, reference model, observer and filters, but also the navigation and how it all can be implemented into a computer.

Chapter 5 presents the improvements and additions made in SAMS. This chapter also gives a short introduction to SAMS and how input and output to the simulator can be processed.

Chapter 6 goes through the basic theory behind Transmission Control Protocol (TCP), and introduces the TCP interface developed during this thesis.

Chapter 7 goes through the implementation of the motion control system implemented in Simulink. In addition selected results from the verification of the performance of the motion control system implemented will be shown in this chapter. Lastly the web application and its implementation is introduced.

Chapter 8 presents the general setup of SAMS used for the simulations in this thesis before the modified vessel model of Oden are presented and discussed.

Chapter 9 presents the simulation performed in level ice with results from empirical ice resistance models. The setup of the simulation are presented before the results are given. Lastly a discussion on the results will be given.

Chapter 10 presents the setup, results and discussion of the Oden Arctic Technology Research Cruise (OATRC) transit simulation, performed in order to validate the simulator against full-scale data from Oden.

Chapter 11 goes through the simulations performed with the acceleration Feedback (FB) controller. First the setup of the simulation and the difference between each simulation is explained, before results are given and discussed based on the performance of the different controllers.

Chapter 12 presents the two major simulations performed for this thesis, this includes a DP simulation with and without IM support. First the setup of the two simulations are presented before the results and a discussion on the DP simulation without IM support are presented. Lastly the results and a discussion on the DP simulation with IM support are given, both in terms of the IM and DP vessel.

Chapter 13 discusses the modifications and improvements implemented in the simulator both in terms of performance and functionality.

Chapter 14 summarizes the thesis before providing suggestions for further work.

Literature Review

2.1 Ice Loads from Level Ice

The vessel-ice interaction can occur in many different ice conditions this can for instance be, but not limited to, slushed ice, broken ice, level ice, ice ridges, and icebergs. Among these, the most ideal ice condition for vessels operating in the Arctic region is level ice interaction, Nguyen et al. (2009). When a vessel moves into a large ice floe, this could be level ice or a large broken ice floe, the forces from the ice acting on the hull will increase as the penetration depth in the ice increases. This will continue to happen until the shear forces in the ice exceeds the maximum shear force of the ice and mechanical failures occurs, Lubbad and Løset (2011). These failure of the ice sheets can happen in a number of different modes; pure crushing, bending, buckling, shearing, splitting, or a mixed mode, i.e. two or more of the failure modes are active at the same time. Nguyen et al. (2009) and Su et al. (2014) described how the process works, the ice floe will first start crushing until the stresses exceeds the tensile strength in the ice, resulting in bending before the ice sheet fails some distance away from the interaction zone. The ice floe is then forced by the advancing ship to rotate parallel to the ships hull where it will be submerged and slide along the hull before it will clear away from the ship, as illustrated by Figure 2.1.

The properties of the ice, the ice thickness, the hull of the vessel and the relative velocity between the vessel and ice determines the failure mode. Lubbad and Løset (2011) claimed that bending failure dominates the other modes, however Su et al. (2014) mentions that in some zones of the ships hull where the slope angles are large (typically at the shoulders and midship) crushing may be the only failure mode, this is adding a significant amount of ice resistance. Lubbad and Løset (2011) also stated that if the

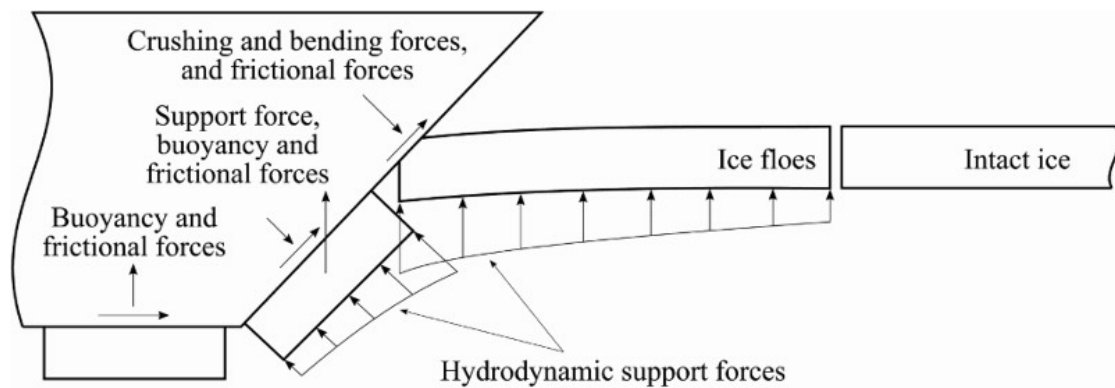


Figure 2.1: Illustration of the ice-hull interaction process. Courtesy of Riska (2010b).

relative velocity is low the vessel-ice interaction will become inelastic. Nguyen et al. (2009) quoted that full-scale operations indicates that the size of the broken off ice floes decreases when the relative velocity between the vessel and ice floe increases, this is confirmed by Tan et al. (2014) and is inline with the physical understanding whereas higher speeds imply higher energy involved. Tan et al. (2014) also stated that the size of the broken floes will increase with higher ice thickness.

Tan et al. (2014) stated that in most cases of ice-breaking the bow of the ship typically opens a channel in the ice wide enough for the entire beam to pass through. Nevertheless situations where the bow shoulders and/or the side of the hull has to force themselves through the ice sheet can occur. This is what is called shoulder crushing, and occurs when there is a relative heading towards the ice sheet. In Su et al. (2014) shoulder crushing is mentioned to be an ineffective way of breaking ice as the hull behind the bow shoulders is in constant contact with the ice edge, building up a significant contribution to the total ice resistance.

2.1.1 Ice Resistance

Erceg and Ehlers (2017) describes ice resistance as:

"... a time average of the instantaneous total forces in longitudinal direction due to ship-ice interaction".

Ice resistance is illustrated in Figure 2.2. There exists a number of semi-empirical ice resistance prediction methods/models, all of them are based on model/full-scale studies and is mostly used in the early ship-design phase. Two of them are the Lindqvist method, Lindqvist (1989), and the Riska method, Riska (1998).

Erceg and Ehlers (2017) stated that the measure of an icebreakers *ice worthiness* is its performance in level ice, they also claims that among other things a proper design of the

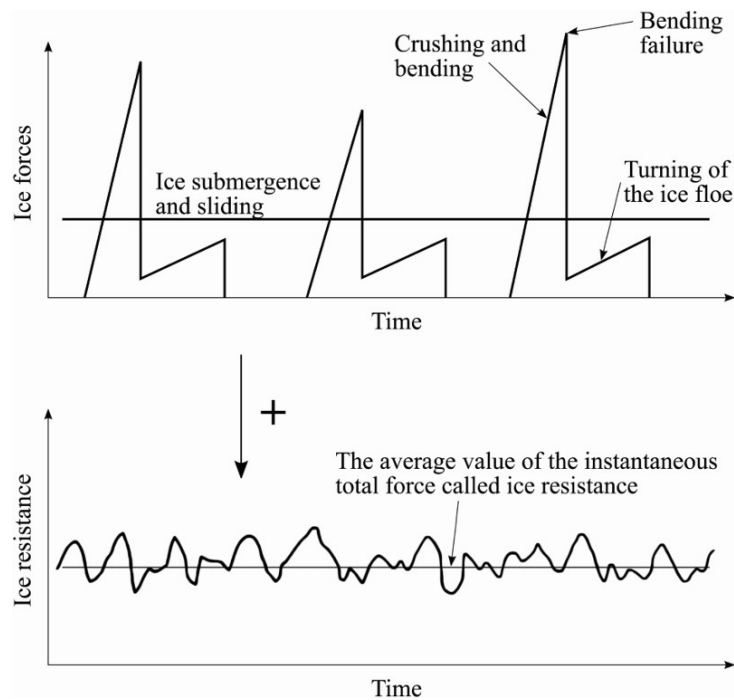


Figure 2.2: Illustration of time history of ice forces contributing to ice resistance defined as an average longitudinal load. Courtesy of Riska (2010a).

ships hull is paramount to the performance as it can reduce the total resistance of the ship. This indicates that a study of different types of hulls is important to assess a ship's overall performance in ice.

A natural sea ice field will contain a mix of intact ice floes, deformed ice floes, ice ridge fragments, open water segments and possibly multi-year ice, Metrikin et al. (2015). Level ice is an idealization of the ice conditions, and despite the unlikelihood of level ice to occur in nature it is considered the standard condition for vessels operating in ice, e.g. for the assignment of ice class, Metrikin et al. (2015). The level ice assumption makes testing in model-scale basins like the Hamburg Ship Model Basin (HSVA) feasible as level ice easily can be created.

There exists a number of semi-empirical methods for estimating a vessel's resistance in level ice, the main difference between the methods being how they were developed. One way is to use a method purely based on full-scale data, e.g. Vance (1980). This method is a result of a purely empirical model and it does not take into account the size or shape of the vessel, but as found by Erceg and Ehlers (2017) the results matched relatively well for vessels with similar sizes. Another model based on full-scale data is presented in Keinonen and Browne (1991) and Keinonen et al. (1996). This model was based on extensive full-scale trials for eighteen of the most important icebreakers at that

time to include the hull shape and size, engine power, environmental conditions etc. into the formula. Another method of developing a model is presented in Riska (1998), this formula was based on three different existing formulas, among these the one presented by Lindqvist (1989). The formula has been expanded and calibrated against full-scale data from ten different Finnish vessels with Baltic ice class to ensure that the model gives reasonable data.

2.1.2 Simulation of Ice Loads

Section 6 in Lubbad and Løset (2011) described a numerical way to calculate the response of the breakable ice, and an estimate on how the force will affect the vessel and how the ice field will break based on these forces. The main assumption is that large ice floes can be modelled as an elastic plate resting on an elastic foundation. This allows the bending stress to be calculated using the power series solution described in, Nevel (1961). Metrikin (2014) continued on the work of Lubbad and Løset (2011) and went further by describing how radial cracks in the ice floe can be modelled. It should be noted that these models are used to calculate realistic stresses in the ice sheet to get more accurate breaking of the ice sheet. Other resistance models such as those mentioned earlier would also need to be implemented to get the correct total resistance on the vessel.

The numerical models mentioned above are used in the respective authors simulators where they simulate vessel-ice interactions. Since the simulator described in Metrikin (2014) was built on the one described in Lubbad and Løset (2011) it is more advanced in the type of patterns the broken of ice takes. Tan et al. (2014) stated that both full-scale and basin-model tests show that the ice edge breaks of in both half-moon shapes (cusps) and in the shape of wedges, as is true in the simulator described in Metrikin (2014).

2.1.3 Ice Drift

A large degree of the sea ice models available are implemented in climate models, and most of these are based on Hibler III (1979) and Girard et al. (2009). The main idea behind the model proposed by Hibler is to couple the ice dynamics with the thickness of the ice, this is done by regulating the ice-ice interactions based on the thickness of the ice. For this to work the ice sheet is considered plastic with a plastic strength depending on both the concentration of ice and the thickness Hibler III (1979). Hunke and Dukowicz (1997) improved the computational efficiency of Hibler's method by assuming an elastic-viscous-plastic model. This model was criticized by Weiss et al. (2007) for a lack of physical basis, but remains in high use due to high computational efficiency.

Weiss et al. (2007) also proposed that when the ice field is looked at as one, fracture- and frictional-sliding are the two main factors in the inelastic deformation. Instead of an elastic-viscous-plastic behaviour this leads to an elastic-brittle behaviour. Other models include Schreyer et al. (2006) which proposed an elastic-decohesive model. Several of these models have been compared and analyzed in Girard et al. (2009), though only on timescales from hours and up.

In Leppäranta (2005) the kinematics of the ice floes was described both as a function of concentration and shape. When the concentration of ice is low, forces are transmitted poorly, on the other hand when concentration is high, the ice floes tend to group together and lose their individual properties as stress increases. The shape will then have big effect where rounded *summer floes* give less total stress and rectangular and pentagonal *winter floes* give a larger transfer of forces, Leppäranta (2005). Leppäranta (2005) also stated that the drift speed usually lies under 1 m/s, but can in some isolated cases, e.g. along coastlines or in narrow canals/rivers reach a higher speed.

2.2 Control Systems

A standard DP control system can be found in for instance Fossen (2011) or Sørensen (2012), but as Jenssen et al. (2009) among others stated, DP controllers for Arctic environments need to be more responsive and track unknown external forces better. Skjetne et al. (2014) discussed two different strategies for DP, namely reactive and proactive control strategy. Kjerstad et al. (2011) presented a method where the acceleration was used as a Feedforward (FF) to model uncertainties. This was done by comparing the filtered acceleration with the acceleration due to known and modeled forces. Kjerstad and Skjetne (2014) used the linear velocity to improve performance, and stated that there is a need for a more aggressive tuning of the observer to get better tracking performance, hence this method requires high quality measurements.

Other more advanced control systems for use in DP also exists, Witkowska (2013) listed a number of model-based controls that have been used for conventional open water DP: Linear Quadratic Gaussian (LQG)-, sliding mode-, robust H-infinity ($H-\infty$)-, nonlinear backstepping-, artificial intelligence-, fuzzy logic- and neural networks-controller.

Du et al. (2015) presented an adaptive robust output FB control that does not require velocity or knowledge of the ships dynamics. This was done by using a high gain observer and a Radial Basis Function (RBF) neural network in vectorial backstepping. Jayasiri et al. (2016) suggested an Unscented Kalman Filter (UKF) nonlinear observer to estimate the vessel's velocity and the unknown ice loads and then use a Nonlinear Model Predictive Controller (NMPC) to get a better disturbance rejection and setpoint

control. The main drawback with using nonlinear control is the increased complexity that makes it harder and more time consuming to tune the controller.

As stated before when a ship enters level ice the loads will be huge and unpredictable and conventional open water DP controllers are not sufficient to counteract these loads, Keinonen et al. (1999) and Moran et al. (2006). Nguyen et al. (2009) proposed a controller where a level ice load model is added to the DP controller to enhance the performance in level ice. Sørensen (2011) described a more realistic scenario where a ship will be exposed to both ice and open water during a operation, hence a switching/hybrid controller is advantageous since it can use a more optimal open water controller in the ice free zones.

2.2.1 Ice Navigation

Choi et al. (2015) stated that an ice navigation system is essential for safe maneuvering and route selection in Arctic regions. This is because choosing the best route by hand is challenging since the ice conditions are changing by the hour, however an ice navigation system is highly dependent on a good IM system and especially a good *ice intelligence* system

One common solution to the problem with large and unpredictable ice forces is to introduce IM. It has been demonstrated by full-scale, model-scale, and numerical experiments that high-uptime positioning is possible with IM support, Kjerstad and Skjetne (2014). According to Eik (2010) IM is defined as:

”the sum of all activities where the objective is to reduce or avoid actions from any kind of ice features”.

IM can typically be divided into two main parts, one involving sea ice management while the other involving iceberg management, Skjetne et al. (2014). One important aspect of sea ice management is ice intelligence.

Ice intelligence is the part of IM which is *”... the process of collecting and analyzing relevant information about the ice environment in a region of interest”*, Haugen et al. (2011). This requires a vast suite of sensors to acquire the needed data whereby satellite images and/or Unmanned Underwater Vehicle (UUV) operating under the ice to measure ice thickness is most common. Haugen et al. (2011) also listed several other sensors with pros and cons for use in ice intelligence. Some examples are:

- *Lasers*: Can be used to measure distance to nearby ice floes, however cannot penetrate dense clouds.

- *Microwave*: Can penetrate clouds and based on the properties of the ice it can distinguish first-year from multi-year ice. The major drawback being the challenges in interpreting the images.
- *Satellites*: The main observation tool for monitoring sea ice, but the high altitude results in a trade of between high resolution and wide coverage range of the sensor data.

When the information about the surrounding region is analyzed an optimal path can be calculated according to the ice thickness or to avoid collisions with possible icebergs. This can for instance be achieved by using state-of-the-art path planning algorithms such as the collision avoidance algorithm described in Loe (2008).

2.3 Historic DP Operations in Ice Infested Waters

Gürtner et al. (2012) claimed that the Arctic holds approximately a quarter of the undiscovered oil and gas in the world. Warmer climate makes these resources more and more available for the offshore industry. The majority of the expected oil and gas deposits in the Arctic are believed to be located beyond the 100 m water depth contour, this is considered to be the upper limit for bottom-founded structures in areas with possible ice contact, Hamilton (2011). Robust floating operations are therefore required, possibly supported by a fleet of IM vessels. Gürtner et al. (2012) also stated that the first DP operation in a moving pack ice was executed already back in 1999 offshore of Sakhalin, this operation is presented in Keinonen et al. (1999). Despite this relatively early start for DP in the Arctic, the oil field developments have very little operational experience regarding DP in ice, Jenssen et al. (2009).

As stated above, only a few DP operations have been performed in icy seas. The well known ones among these were the Sakhalin diving operation in spring of 1999, Keinonen et al. (1999), where the ice condition exceeded 9/10th (90 %) ice coverage, including level ice, ridges, and large ice floes. The DP vessel was assisted by two ice-breakers and the downtime for the DP vessel was 22 %. The Arctic Coring Expedition (ACEX) at the Lomonosov ridge in August 2004 presented in Moran et al. (2006) is another DP operation, here the ice conditions were even more severe than in Sakhalin, and the DP operations could only be achieved by manual control, giving a downtime of 38 % for the DP vessel. The coring operation in Northeast Greenland in 2008, Rohlén (2009), is another example of Arctic DP operations.

Wright (1999) reported the results of a Program of Energy Research and Development

(PERD) sponsored study on behalf of Natural Resources Canada (NRCan)¹. The goal of the study was to explore the loads on a moored vessel in moving pack ice, this was done on the Grand Banks outside the eastern coast of Canada. This study focused on documenting the loads on the drilling vessels Kulluk and Canmar, and found that they can maintain position in moderate to relatively high ice concentration with a high frequency of change in the ice drift direction, giving a downtime of less than 10 %. However that required extensive use of IM support, requiring up to three icebreakers during the worst conditions.

2.4 Simulation of Vessels in Ice Infested Waters

Simulation of maneuvering and station keeping in ice fields are relatively new, this is partly due to the high demands in computational power and the higher availability of Arctic waters due to climate change in recent years. This field only recently started to be explored fully (around year 2010). Despite the late start to this field the first NIT simulator, according to Metrikin (2014), was proposed by Valanto and Puntigliano (1997). This simulator was used to simulate the ice resistance of a vessel in level ice doing ice-breaking maneuvers. Later a fully coupled NIT environment was described by Derradji-Aouat (2010). This simulator was based on Explicit Finite Element Method (EFEM) implemented within the commercial packages ANSYS² and LS-DYNA³. This technique was used among others by Wang and Derradji-Aouat (2010) to simulate both model-scale and full-scale interactions of structures with broken ice. Lee et al. (2013) proposed a new NIT simulator to simulate level ice, this simulator was based on the multi-material arbitrary Lagrangian formulation. The fluid-structure interaction in this simulator was analyzed by use of the already existing LS-DYNA code.

Metrikin (2014) described a numerical simulator for station keeping in sea ice. Janßen et al. (2017) has gone further and proposed a method for using Graphics Processing Unit (GPU) for faster processing, this is though limited to rigid ice so can only be used after the sea ice has been broken. This paper also explored more into the hull design and how this can be made to keep ice away from the propulsion devices. The Efficient Lattice Boltzman Environment (ELBE) solver Janßen (2016) is a Computational Fluid Dynamic (CFD) solver that has been used for simulating structure-ice interactions. In level ice Lubbad and Løset (2011) has developed a method for real-time simulation of structure-ice interactions, this method has been further developed into a NIT simulator where Denk (2016) has made a DP control system using Python and an external com-

¹<http://www.nrcan.gc.ca/home>

²<http://www.ansys.com/>

³<http://www.lstc.com/>

munication protocol. Metrikin et al. (2012) explored the possibilities to use different physics engines to better handle the increased computational complexity of simulating larger ice floes. Metrikin et al. (2013) discussed some methods for simulating ice loads on a DP vessel, these methods are in short:

- *Empirical and statistical models*: These models are based on full-scale and model-scale observations, and then express the loads as a function depending on the ice conditions and the vessel.
- *Experimental data method*: This method is based on actual experimental data which can be obtained from both full-scale and model-scale tests. This data is then used as direct input to a virtual DP system simulator, so called Hardware In The Loop (HIL) simulation.
- *Physically based simulation*: These are built on the fundamental laws of physics, and will simulate the system dynamics realistically, however these simulations have a high complexity which in turn increase simulation time.

The study performed by PERD described in Wright (1999) did a comparison of their full-scale results to model testing and found that the results of the model testing loads are ”... *surprisingly good, for equivalent ice interaction situations*”. It is worth noting that the loads tends to lie towards the upper end of the full-scale results, however compared to full-scale and model-scale tests of other vessels this is not uncommon. This shows that it is possible to validate a NIT simulator against model-scale data, making the validation process considerably easier and more cost efficient. With this the simulator has an advantage over model-scale tests as it can be modified and tested with different vessels and ice conditions in a faster and more cost efficient way.

2.5 Digital Twins

Digital twins as a concept can according to Zhang et al. (2017) be seen as:

”... one of the approaches to achieving the interconnection between physical world and digital world”.

In short a digital twin can be seen as a digital equivalent of a physical system, and they are highly useful in many different industries such as space-, aerospace-, automobile- and maritime-industry. Digital twins can be used in an early design phase where they can uncover costly errors before the physical system is built. Digital twins can also be used in simulations for training crews and used to monitor a physical system for wear and tear. Another advantage to the digital twins is that they can be modeled completely,

i.e. they contain the whole system with all its subsystems. Smogeli (2017) stated that *"you cannot verify key properties such as safety, reliability or availability as a paper exercise only, or by analyzing individual components and subsystems and then aggregating results"*, this is also backed by Glaessgen and Stargel (2012). Meaning that for a complete verification of a system, a test with the whole system and its subsystems is needed.

Although the idea of a digital twin is not something new, the term digital twin was first introduced at a presentation in 2002, Grieves and Vickers (2016). Glaessgen and Stargel (2012) stated that a paradigm shift is needed to improve the *"shortcoming of conventional approaches"*. This is what has been happening in the last few years with new, better and cheaper technologies, new sensors giving a better connection between the digital and physical world, and new computer hardware making it possible to speed up simulations of these digital twins. The quote above by Glaessgen and Stargel (2012) is one of many stating the significance digital twins has in the different industries. Glaessgen and Stargel (2012) also gave examples as to why the paradigm shift is needed, one example is that it can be difficult to reproduce the extreme loads systems will endure in a laboratory. Another example is that future systems will be pushed even further in regards to physical strain, temperatures, pressure and other parameters, again stating the importance of full system simulations.

Digital twins are also more and more used in the maritime industry, as an example HIL-testing is testing a control system on a digital twin incorporating multiple subsystems. Smogeli (2017) stated that Det Norske Veritas (DNV GL) has done this on several ships and that it can easily be adapted for any ship type. Smogeli (2017) also stated that the movement to more cyber-physical systems such as autonomous vessels needs to be completely verified as a whole, before the system can be deployed.

Ice Forces and Vessel Dynamics

In this chapter five different ice resistance models will be presented as well as how ice forces that acts on the vessel can be estimated. Moving on the dynamics of a vessel will be presented, this includes both the kinematics and the kinetics of the vessel together with both hydro-static and -dynamic forces. At last a summary of the vessel dynamics is presented as one unified model.

3.1 Ice Resistance Models

Ice forces can become quite large, and with few exceptions ice forces are the major concern when designing marine structures for the Arctic, Palmer and Croasdale (2014). As limited data and lack of theory on structures operating in ice, the engineers responsible for designing these structures/vessels have to rely on empirical and semi-empirical models for estimating the ice loads the structures will endure. These models are based on field tests, small-scale tests, theory or a combination of these, Palmer and Croasdale (2014).

3.1.1 Level ice

The level ice resistance is typically divided into three components, breaking including both crushing and bending, submersion with sliding and a velocity dependent component. The first two terms are in line with the physical understanding of the interaction while for the last term, empirical coefficients have been developed for the models, Erceg and Ehlers (2017). In the following four different level ice resistance prediction models are briefly presented.

3.1.1.1 Vance (1980)

The model presented in Vance (1980) is solely based on one full-scale data set for the USCG Katmai Bay. As this model is only developed from one data set on a relatively small vessel, care should be taken when using it, however Erceg and Ehlers (2017) reported that the model matches a significantly larger ship, the MT Sotka. The model predicts the ice resistance to be

$$R_{ice} = 55.8583(\rho_{water} - \rho_{ice})gBh^2 + 0.0188\sigma_fVB\sqrt{\frac{h}{g}}. \quad (3.1)$$

Here the two terms can be seen as submersion and velocity dependent, the submersion part of the equation varies with the square of the ice thickness, where the velocity dependent part varies with the flexural strength and the square root of the ice thickness.

3.1.1.2 Lindqvist (1989)

The approach described in Lindqvist (1989) was to identify the main components in the ice resistance and derive a formula from these. Lindqvist also added the dimensions of the vessel, hull shape and ice properties to show how these affect the ice resistance, the formulae were then corrected against full-scale data for different vessels. Erceg and Ehlers (2017) stated that the results are reliable for larger vessels, however for smaller vessels the velocity dependent part of the resistance is found unreliable, but in the right order of magnitude. The resistance model is given in (3.2) and consists of bow crushing R_c , bending R_b , submersion R_s and a velocity component. Formulas for calculating the resistance components R_c , R_b and R_s are given in Appendix B.1. Note that in order to shorten the calculations Lindqvist approximated the underside of the vessel as a flat surface and both the deflection of the ice and trim of the vessel are neglected.

$$R_{ice} = (R_c + R_b)\left(1 + \frac{1.4V}{\sqrt{gh}}\right) + R_s\left(1 + \frac{9.4V}{\sqrt{gL}}\right). \quad (3.2)$$

3.1.1.3 Keinonen and Browne (1991); Keinonen et al. (1996)

The model described in Keinonen and Browne (1991) and Keinonen et al. (1996) was derived from extensive research of eighteen of the most modern icebreakers in the world at that time. Hence this model are covering the full spectrum of icebreakers design features, engine power, displacement etc. This gives the model an advantage given the broad spectrum of data of which it has been formulated upon, and this model is the most used model nowadays. The total resistance is given in (3.3) and is divided

into three components, the first term R_{ow} is the open water resistance, the second term $R(1 \text{ m/s})_{ice}$ is the level ice resistance at 1 m/s and the third term $R(> 1 \text{ m/s})_{ice}$ is the level ice resistance above 1 m/s. The last term can be considered a velocity dependent term of the total resistance, the fully expanded formulas are given in Appendix B.2.

Due to different bow form which can effect the ice resistance the model has divided the formulas into two groups based on the bow form. The first group is vessels with sharp chined hulls, i.e. having a fairly consistent angle across the beam, and the second group is vessels with rounded hull forms, i.e. having a gradual change in angles between stem and shoulders.

$$R(V)_{total} = R(V)_{ow} + R(1 \text{ m/s})_{ice} + R(> 1 \text{ m/s})_{ice}. \quad (3.3)$$

3.1.1.4 Riska (1998)

The model described in Riska (1998) is a simplified prediction model which is based on three other resistance models, among these is the above described Lindqvist model. The model described by Riska is also calibrated against full-scale data from ten Baltic-classed vessels. For convenience the model is simplified and the thickness of the ice sheet are the only ice related variable to affect the resistance. As seen in (3.4) the model is in two parts, one velocity dependent and one velocity independent, the full model is described in Appendix B.3.

$$R_{ice} = C_1 + C_2V. \quad (3.4)$$

3.1.2 Managed Ice Loads

In reality most vessels and structures designed for the Arctic, and especially floaters, are designed to operate in a managed ice fields to reduce the peak load it has to sustain. Palmer and Croasdale (2014) proposed an empirical model for calculating the ice load in surge direction in managed ice to be

$$R_{ice} = p_a B h \left(1 + \frac{\mu_i}{\tan \beta} \right) + \frac{p_c B h}{\tan \beta} + 2Lh(\mu_i p_a + p_c). \quad (3.5)$$

Here p_a is the ambient ice pressure, p_c is the ice field cohesion and β is the angle of potential ice accumulation at the bow of the vessel. B , L and h are the same as for the previous models, namely beam, length and ice thickness respectively.

3.2 Ice Force Estimation

Kjerstad et al. (2018) stated that there are two methods of estimating the ice load on a vessel, namely a direct and an indirect method. The indirect method is to do ice environment monitoring together with ice resistance models, e.g. Riska (1998) for level ice or Palmer and Croasdale (2014) for managed ice. However it can be difficult to quantify the surrounding ice field in real-time and this method is also limited by the ice resistance models. The direct method is to measure the resistance directly on the vessel, and this can be achieved in two ways. Measuring the local ice loads acting on the hull by measuring the strain and deflections with strain gauges, or by measuring the global load from accelerometers and/or Inertial Measurement Units (IMUs). The major drawback to the estimation of local ice loads as stated by Kjerstad et al. (2018) is the costly installation and calibration of the strain gauges, and some places on the hull they can be hard or impossible to install. Note that when estimating the local force on the hull the global force is just a summation of all the measurements.

The general idea when using accelerometers for estimating global ice loads is to measure the general movement of the vessel, and by then knowing the thruster inputs, wind velocity and vessel velocity, well established models, see Section 3.7 and (Fossen, 2011, chap. 8), can be used in the calculation of the global ice loads. Kjerstad et al. (2018) proposed an algorithm using four different IMUs setup such that they form a virtual accelerometer in Origin of $\{b\}$ (CO), this also makes it possible to measure the angular acceleration of the vessel by knowing the relative positions between the accelerometers.

3.3 Model Kinematics

The kinematics considers only the geometrical aspects of motion, Fossen (2011). In this thesis the notation of kinematics will follow SNAME (1950) as given in Table 3.1, note that this is only valid for the BODY-fixed reference-frame. The following two different coordinate systems will be used; North-East-Down (NED) $\{n\} = (x_n, y_n, z_n)$ and BODY $\{b\} = (x_b, y_b, z_b)$. The NED coordinate frame is chosen to coincide with the inertial reference-frame in this thesis.

From Table 3.1 $\mathbf{v}_{b/n}^b$, $\boldsymbol{\omega}_{b/n}^b$, \mathbf{f}_b^b and \mathbf{m}_b^b can be defined. $\mathbf{v}_{b/n}^b$ is the BODY-fixed linear velocity with respect to $\{n\}$ given in $\{b\}$, $\boldsymbol{\omega}_{b/n}^b$ is the BODY-fixed angular velocity with respect to $\{n\}$ given in $\{b\}$, \mathbf{f}_b^b is the BODY-fixed linear force acting through o_b and \mathbf{m}_b^b the BODY-fixed moment acting through o_b , with o_b being the origin of $\{b\}$, i.e. CO. In addition define $\mathbf{p}_{b/n}^n = [x_n, y_n, z_n]^\top$ as the position of the BODY reference-frame with respect to $\{n\}$ given in $\{n\}$.

	Forces and moments	Linear and angular velocities	Positions and Euler angles
Motion in x direction (surge)	X	u	x
Motion in y direction (sway)	Y	v	y
Motion in z direction (heave)	Z	w	z
Rotation about the x axis (roll)	K	p	ϕ
Rotation about the y axis (pitch)	M	q	θ
Rotation about the z axis (yaw)	N	r	ψ

Table 3.1: SNAME notation.

To convert one coordinate frame to another a rotation matrix must be defined. A rotation matrix with rotation β about an axis $\boldsymbol{\lambda}$ where $\|\boldsymbol{\lambda}\| = 1$ can be defined from Fossen (2011) as

$$\mathbf{R}_{\boldsymbol{\lambda},\beta} = \mathbf{I}_{3 \times 3} + \sin(\beta)\mathbf{S}(\boldsymbol{\lambda}) + [1 - \cos(\beta)]\mathbf{S}^2(\boldsymbol{\lambda}), \quad (3.6)$$

where $\mathbf{S} = -\mathbf{S}^\top$ is a skew-symmetric matrix. Inserting $\boldsymbol{\lambda} = [1, 0, 0]^\top$, $\boldsymbol{\lambda} = [0, 1, 0]^\top$ and $\boldsymbol{\lambda} = [0, 0, 1]^\top$ for the x , y and z axis respectively with $\beta = \phi$, $\beta = \theta$ and $\beta = \psi$ respectively into (3.6) gives

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}, \mathbf{R}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}, \mathbf{R}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.7)$$

where $c\cdot = \cos(\cdot)$ and $s\cdot = \sin(\cdot)$.

Defining the Euler angles as $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^\top$, where ϕ , θ and ψ are the Euler angles between NED and BODY. Then by using a zyx convention of rotation $\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$ will be defined as the transform from BODY to NED using the Euler angles. This is equivalent to multiplying $\mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{x,\phi}$ from (3.7), and expanding yields

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - c\phi s\psi & s\psi s\phi + c\psi c\phi s\theta \\ c\theta s\psi & c\psi c\phi + s\psi s\theta s\phi & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}. \quad (3.8)$$

In many situations regarding surface vessels only the horizontal modes of motion are of

interest, hence (3.8) can be reduced by assuming small roll (ϕ) and pitch (θ) angles to the 3 Degree of Freedom (DOF) rotation matrix in the horizontal plane,

$$\mathbf{R}_b^n(\Theta_{nb}) \approx \mathbf{R}_{z,\psi} \cdot \mathbf{I}_{3 \times 3} \cdot \mathbf{I}_{3 \times 3} = \mathbf{R}_b^n(\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ c\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.9)$$

The transpose of the matrices in (3.8) and (3.9) will give the transformation from NED to BODY, i.e. $\mathbf{R}_n^b = (\mathbf{R}_b^n)^\top$. For the rest of this thesis the rotation matrix \mathbf{R} will be the rotation matrix from BODY to NED, and \mathbf{R}^T will be the rotation from NED to BODY.

The Euler angles can be obtained from quaternions, $\mathbf{q} = (\eta, \epsilon_1, \epsilon_2, \epsilon_3)$, by the following formula given in Fossen (2011),

$$\begin{aligned} \phi &= \text{atan2}(2(\eta\epsilon_1 + \epsilon_2\epsilon_3), 1 - 2(\epsilon_1^2 + \epsilon_2^2)), \\ \theta &= \text{asin}(2(\eta\epsilon_2 - \epsilon_3\epsilon_1)), \\ \psi &= \text{atan2}(2(\eta\epsilon_3 + \epsilon_1\epsilon_2), 1 - 2(\epsilon_2^2 + \epsilon_3^2)), \end{aligned} \quad (3.10)$$

where $\text{asin}(\cdot) = 1/\sin(\cdot)$, $\text{atan2}(a, b)$ represents the four quadrant $1/\tan(a/b)$ and $\theta \neq \pm\pi/2$. Caution should be taken if $\theta \rightarrow \pm\pi/2$, however this will not be a problem for surface going vessels.

Next define the position and orientation vector of a vessel as

$$\boldsymbol{\eta} := \begin{bmatrix} \mathbf{p}_{b/n}^n \\ \Theta_{nb} \end{bmatrix}. \quad (3.11)$$

Furthermore define the velocities and forces acting on the vessel as

$$\boldsymbol{\nu} := \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix}, \boldsymbol{\tau} := \begin{bmatrix} \mathbf{f}_b^b \\ \mathbf{m}_b^b \end{bmatrix}. \quad (3.12)$$

Using this a new matrix $\mathbf{J}_\Theta(\boldsymbol{\eta})$ can be defined as

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} = \mathbf{J}_\Theta(\boldsymbol{\eta}) \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix}. \quad (3.13)$$

It is then trivial to see that

$$\dot{\mathbf{p}}_{b/n}^n = \mathbf{v}_{b/n}^n = \mathbf{R}_b^n(\Theta_{nb})\mathbf{v}_{b/n}^b. \quad (3.14)$$

Together with the realization that the Euler rate $\dot{\Theta}_{nb}$ can only depend on the angular velocity $\omega_{b/n}^b$, (3.13) becomes

$$\dot{\eta} = \begin{bmatrix} \dot{p}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} \begin{bmatrix} \mathbf{R}_b^n(\Theta_{nb}) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_\Theta(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \omega_{b/n}^b \end{bmatrix}. \quad (3.15)$$

From Fossen (2011) the matrix $\mathbf{T}_\Theta(\Theta_{nb})$ is given as

$$\mathbf{T}_\Theta(\Theta_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}, \quad (3.16)$$

where $c \cdot = \cos(\cdot)$, $s \cdot = \sin(\cdot)$ and $t \cdot = \tan(\cdot)$. For small (δ) roll and pitch movements the expression can be simplified to

$$\mathbf{T}_\Theta(\Theta_{nb}) \approx \begin{bmatrix} 1 & 0 & \delta\theta \\ 0 & 1 & -\delta\phi \\ 0 & \delta\phi & 1 \end{bmatrix} \approx \mathbf{I}_{3 \times 3}. \quad (3.17)$$

A 3DOF model of (3.15) can then be simplified by using (3.9) and (3.17) to

$$\dot{\eta} = \mathbf{R}(\psi)\boldsymbol{\nu}, \quad (3.18)$$

where $\boldsymbol{\eta} = [x_n, y_n, \psi]^\top$ and $\boldsymbol{\nu} = [u, v, r]^\top$.

3.4 Model Kinetics

Kinetics or dynamics is concerned with the effect of linear forces and moments acting on a rigid body. From Fossen (2011) one can see that from Euler's second axiom a rotation motion about the vessels Center of Gravity (CG) can be expressed as

$$\mathbf{m}_g^b = \mathbf{I}_g \dot{\omega}_{b/n}^b - \mathbf{S}(\mathbf{I}_g \omega_{b/n}^n) \omega_{b/n}^n, \quad (3.19)$$

where \mathbf{S} is a skew-symmetric matrix as defined in section 3.3, \mathbf{m}_g^b is the moment acting through CG given in $\{b\}$ and \mathbf{I}_g is the inertia matrix about CG defined as

$$\mathbf{I}_g := \begin{bmatrix} \mathbf{I}_{xx} & -\mathbf{I}_{xy} & -\mathbf{I}_{xz} \\ -\mathbf{I}_{yx} & \mathbf{I}_{yy} & -\mathbf{I}_{yz} \\ -\mathbf{I}_{zx} & -\mathbf{I}_{zy} & \mathbf{I}_{zz} \end{bmatrix}. \quad (3.20)$$

From Euler's first axiom a translation motion can be found in Fossen (2011) to be

$$\mathbf{f}_g^b = m[\dot{\mathbf{v}}_{g/n}^b + \mathbf{S}(\boldsymbol{\omega}_{b/n}^b)\mathbf{v}_{g/n}^b], \quad (3.21)$$

where \mathbf{f}_g^b is the force acting through CG given in $\{b\}$ and $\mathbf{v}_{g/n}^b$ is the velocity in CG with respect to $\{n\}$ given in $\{b\}$. Combining (3.19) and (3.21) gives

$$\begin{bmatrix} \mathbf{f}_g^b \\ \mathbf{m}_g^b \end{bmatrix} = \begin{bmatrix} m\mathbf{S}(\boldsymbol{\omega}_{b/n}^b) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\mathbf{S}(\mathbf{I}_g\boldsymbol{\omega}_{b/n}^b) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{g/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} + \begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_g \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_{g/n}^b \\ \dot{\boldsymbol{\omega}}_{b/n}^b \end{bmatrix}. \quad (3.22)$$

By then defining \mathbf{C}_{RB} as the rigid body Coriolis and centripetal matrix due to rotation and \mathbf{M}_{RB} as the rigid body inertia matrix (3.22) can be expressed as

$$\begin{bmatrix} \mathbf{f}_g^b \\ \mathbf{m}_g^b \end{bmatrix} = \mathbf{C}_{RB}^{CG} \begin{bmatrix} \mathbf{v}_{g/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} + \mathbf{M}_{RB}^{CG} \begin{bmatrix} \dot{\mathbf{v}}_{g/n}^b \\ \dot{\boldsymbol{\omega}}_{b/n}^b \end{bmatrix}. \quad (3.23)$$

For most vessels it is advantageous to derive the motion equations from CO rather than CG. This can be achieved by a coordinate transform from CG to CO given in Fossen (2011) as

$$\mathbf{v}_{g/n}^b = \mathbf{v}_{b/n}^b + \mathbf{S}^\top(\mathbf{r}_g^b)\boldsymbol{\omega}_{b/n}^b, \quad (3.24)$$

where \mathbf{r}_g^b is a vector denoting the distance from CO to CG given in $\{b\}$. Noting that this equation do not depend on angular velocity a transform from CG to CO can be defined by the use of a matrix $\mathbf{H}(\mathbf{r}_g^b)$ as

$$\begin{bmatrix} \mathbf{v}_{g/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} = \mathbf{H}(\mathbf{r}_g^b) \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix}. \quad (3.25)$$

It is then trivial to see by using (3.24) and (3.25) that $\mathbf{H}(\mathbf{r}_g^b)$ becomes

$$\mathbf{H}(\mathbf{r}_g^b) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{S}^\top(\mathbf{r}_g^b) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}. \quad (3.26)$$

Furthermore using (3.25) on (3.23) gives

$$\mathbf{H}^\top(\mathbf{r}_g^b) \begin{bmatrix} \mathbf{f}_g^b \\ \mathbf{m}_g^b \end{bmatrix} = \mathbf{H}^\top(\mathbf{r}_g^b) \mathbf{C}_{RB}^{CG} \mathbf{H}(\mathbf{r}_g^b) \begin{bmatrix} \mathbf{v}_{g/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} + \mathbf{H}^\top(\mathbf{r}_g^b) \mathbf{M}_{RB}^{CG} \mathbf{H}(\mathbf{r}_g^b) \begin{bmatrix} \dot{\mathbf{v}}_{g/n}^b \\ \dot{\boldsymbol{\omega}}_{b/n}^b \end{bmatrix}. \quad (3.27)$$

The Coriolis and centripetal matrix and the mass matrix in CO can then be defined as

$$\mathbf{C}_{RB}^{CO} := \mathbf{H}^\top(\mathbf{r}_g^b) \mathbf{C}_{RB}^{CG} \mathbf{H}(\mathbf{r}_g^b), \quad (3.28)$$

$$\mathbf{M}_{RB}^{CO} := \mathbf{H}^\top(\mathbf{r}_g^b) \mathbf{M}_{RB}^{CG} \mathbf{H}(\mathbf{r}_g^b). \quad (3.29)$$

Expanding (3.28) and (3.29) gives

$$\mathbf{C}_{RB}^{CO} = \begin{bmatrix} m\mathbf{S}(\boldsymbol{\omega}_{b/n}^b) & -m\mathbf{S}(\boldsymbol{\omega}_{b/n}^b)\mathbf{S}(\mathbf{r}_g^b) \\ m\mathbf{S}(\mathbf{r}_g^b)\mathbf{S}(\boldsymbol{\omega}_{b/n}^b) & \mathbf{S}((\mathbf{I}_g - m\mathbf{S}^2(\mathbf{r}_g^b))\boldsymbol{\omega}_{b/n}^b) \end{bmatrix}, \quad (3.30)$$

$$\mathbf{M}_{RB}^{CO} = \begin{bmatrix} m\mathbf{I}_{3 \times 3} & -m\mathbf{S}(\mathbf{r}_g^b) \\ m\mathbf{S}(\mathbf{r}_g^b) & \mathbf{I}_g - m\mathbf{S}^2(\mathbf{r}_g^b) \end{bmatrix}, \quad (3.31)$$

noting that $-\mathbf{S}(\cdot)\mathbf{S}^\top(\cdot) = \mathbf{S}(\cdot)\mathbf{S}(\cdot) = \mathbf{S}^2(\cdot)$. This can further be simplified by defining \mathbf{I}_b to be the inertia matrix around $\{b\}$, then by the *parallel axis theorem*, Egeland (2002), and by applying the definition of a skew-symmetric matrix,

$$\mathbf{I}_b = \mathbf{I}_g - m\mathbf{S}^2(\mathbf{r}_g^b). \quad (3.32)$$

Furthermore by noting that $\mathbf{I}_b = \mathbf{I}_b^\top$ and $-\mathbf{S}(\cdot) = \mathbf{S}^\top(\cdot)$ it is trivial to see that $\mathbf{M}_{RB}^{CO} = \mathbf{M}_{RB}^{CO\top}$. Another property can be seen by noting that $\det(\mathbf{S}) \geq 0$ and $\det(\mathbf{I}_b) > 0$ such that $\det(\mathbf{M}_{RB}^{CO}) > 0$. By expanding the equation it is also easy to see that $\dot{\mathbf{M}}_{RB}^{CO} = \mathbf{0}_{6 \times 6}$. From this point onward \mathbf{M}_{RB} will be used as \mathbf{M}_{RB}^{CO} and \mathbf{C}_{RB} will be used as \mathbf{C}_{RB}^{CO} . By using (3.12) the expression can be simplified to, Fossen and Sagatun (1991),

$$\boldsymbol{\tau} = \mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}. \quad (3.33)$$

Note that the mass matrix \mathbf{M}_{RB} is unique for a vessel while there exists several representations of \mathbf{C}_{RB} , and more information on the different representations of \mathbf{C}_{RB} can be found in Fossen and Sagatun (1991), Fossen and Fjellstad (1995) and Fossen (2011).

A simplification of \mathbf{M}_{RB} can be obtained if $\text{CO} = \text{CG}$, then $\mathbf{r}_g^b = \mathbf{0}$ and \mathbf{M}_{RB} becomes

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_b \end{bmatrix}. \quad (3.34)$$

3.5 Hydrostatics

The hydrostatic forces for a surface vessel comprises buoyancy and gravitational forces and are called restoring forces. Ballast systems and their configuration is also a part of hydrostatics, but for this thesis it will be left out, together with computation of natural periods which are assumed known. For this section some of the important variables are Metacentric height (\overline{GM}_i) where $i \in \{T, L\}$ is the transverse- and longitudinal-metacentric height respectively, Water plane area (A_{wp}) and Center of Buoyancy (CB). \overline{GM}_i is defined as the the distance between CG and Metacenter (M_i) where M_i is defined according to Fossen (2011) as:

”The theoretical point at which an imaginary vertical line breaks through CB intersects another imaginary vertical line through a new CB created when the body is displaced, or tilted, in the water.”

Archimedes’ principle states that the buoyancy of an object at rest is equal to the weight of the fluid displaced by the object, this can in mathematical terms be written as

$$mg = \rho g \Delta, \quad (3.35)$$

noting that mg is equal to the buoyancy of the object when the object is at rest, and Δ is the displaced water volume. Now let $z = 0$ denote the equilibrium heave position at rest, then the hydrostatic force in heave Z can be denoted as

$$Z = -\rho g \delta \Delta(z). \quad (3.36)$$

This can be simplified by assuming A_{wp} constant for small heave motions to

$$Z \approx -\rho g A_{wp}(0)z. \quad (3.37)$$

With this the restoring force can be written as

$$\delta \mathbf{f}_r^b \approx \mathbf{R}^\top \delta \mathbf{f}_r^n = \mathbf{R}^\top \begin{bmatrix} 0 \\ 0 \\ -\rho g A_{wp}(0)z \end{bmatrix} = -\rho g A_{wp}(0)z \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{bmatrix}. \quad (3.38)$$

The buoyancy force can be written using (3.35) as

$$\mathbf{f}_r^b = \mathbf{R}^\top \begin{bmatrix} 0 \\ 0 \\ -\rho g \Delta \end{bmatrix} = -\rho g \Delta \begin{bmatrix} -\sin(\theta) \\ \cos(\theta) \sin(\phi) \\ \cos(\theta) \cos(\phi) \end{bmatrix}. \quad (3.39)$$

Furthermore from Fossen (2011) the moment arm in pitch and roll is given as

$$\mathbf{r}_r^b = \begin{bmatrix} -\overline{GM}_L \sin(\theta) \\ \overline{GM}_T \sin(\phi) \\ 0 \end{bmatrix}. \quad (3.40)$$

Now by assuming small contribution and thus neglecting the term from $\delta \mathbf{f}_r^b$ to the moment \mathbf{m}_r^b becomes

$$\mathbf{m}_r^b = \mathbf{r}_r^b \times \mathbf{f}_r^b = -\rho g \Delta \begin{bmatrix} \overline{GM}_T \sin(\phi) \cos(\theta) \cos(\phi) \\ \overline{GM}_L \sin(\theta) \cos(\theta) \cos(\phi) \\ (\overline{GM}_T - \overline{GM}_L \cos(\theta)) \sin(\theta) \sin(\phi) \end{bmatrix}. \quad (3.41)$$

Now let $\mathbf{g}(\boldsymbol{\eta})$ denote the restoring forces, then

$$\mathbf{g}(\boldsymbol{\eta}) = - \begin{bmatrix} \delta \mathbf{f}_r^b \\ \mathbf{m}_r^b \end{bmatrix}. \quad (3.42)$$

Further simplifications can be made by assuming small ϕ , θ and z movements and yz -plane symmetry of the vessel. Then $\cos(\cdot) \approx 1$ and $\sin(\cdot) \approx \cdot$, where \cdot is ϕ and θ . Using this in (3.38) and (3.41) and putting into (3.42) gives

$$\mathbf{g}(\boldsymbol{\eta}) \approx \begin{bmatrix} -\rho g A_{\omega p}(0) z \theta \\ \rho g A_{\omega p}(0) z \phi \\ \rho g A_{\omega p}(0) z \\ \rho g \Delta \overline{GM}_T \phi \\ \rho g \Delta \overline{GM}_L \theta \\ \rho g \Delta (\overline{GM}_T - \overline{GM}_L) \theta \phi \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ \rho g A_{\omega p}(0) z \\ \rho g \Delta \overline{GM}_T \phi \\ \rho g \Delta \overline{GM}_L \theta \\ 0 \end{bmatrix}. \quad (3.43)$$

A linear approximation can be made for convenience such that

$$\mathbf{g}(\boldsymbol{\eta}) \approx \mathbf{G} \boldsymbol{\eta}. \quad (3.44)$$

Then by comparison of (3.43) \mathbf{G} becomes

$$\mathbf{G} = \text{diag}\{0, 0, \rho g A_{\omega p}(0), \rho g \Delta \overline{GM}_T, \rho g \Delta \overline{GM}_L, 0\}. \quad (3.45)$$

3.6 Hydrodynamics

This section will deal with the hydrodynamic forces acting on a vessel in motion. Firstly a short introduction to the potential theory is presented before a hydrodynamic model will be presented. For this section wave forces have been neglected as these forces are not relevant for this thesis. This means that what Fossen (2011) referred to as seakeeping theory (maneuvering in presence of waves) will not be relevant, however fluid memory effects will be explained. At last maneuvering theory which is maneuvering at zero-frequency waves will be explained. All the formulas will be given for a 3DOF model.

3.6.1 Potential Theory

To compute the hydrodynamics correctly, the fluid flow velocity vector needs to be defined as

$$\mathbf{v}(\mathbf{x}, t) = [v_1(\mathbf{x}, t), v_2(\mathbf{x}, t), v_3(\mathbf{x}, t)]^\top, \quad (3.46)$$

where $\mathbf{x} = [x_1, x_2, x_3]^\top$ is a location in the fluid. Then by assuming the vessels hull is continuous, the flow is incompressible and the fluid flow continuous the continuity equation, Fossen (2011), becomes

$$\text{div}(\mathbf{v}) = \nabla \cdot \mathbf{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} + \frac{\partial v_3}{\partial z}. \quad (3.47)$$

Furthermore the Navier-Stokes equation can be found in for instant Çengel and Cimbala (2013) as

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \rho \mathbf{F} - \delta p(\mathbf{x}, t) + \mu \nabla^2 \mathbf{v}, \quad (3.48)$$

where \mathbf{F} is the force due to gravity, $p(\mathbf{x}, t)$ is the pressure and μ is the viscosity coefficient for the fluid. Solving (3.47) and (3.48) together can only be done numerically, therefore further simplifications must be done. Assuming an ideal fluid, i.e. no viscosity, irrotational flow and a scalar function $\Phi = \Phi(t, x, y, z)$ that satisfy $\mathbf{v} = \nabla \Phi$, called a potential. Then the pressure can be found by integrating the Navier-Stokes equation without the viscous term, this is often referred to as Euler's equation, and get

the Bernoulli equation, from for example Çengel and Cimbala (2013), as

$$\frac{p}{\rho} + \frac{\partial\Phi}{\partial t} + \frac{1}{2}(\nabla\Phi)^2 + gz = C, \quad (3.49)$$

where C is a constant. Putting $C = p/\rho$ gives the relative pressure. Noting that the potential satisfy the Laplace equation and that the Laplace equation is linear, Fossen (2011), the problem can be solved in frequency domain. When the potential is solved this can be put into the Bernoulli equation and integrated across the surface to get the hydrodynamic forces.

3.6.2 Seakeeping Theory

As wave forces are not relevant these will not be presented, however when a vessel is moving in water the motion itself creates waves that dissipates energy from the vessel, this is called *fluid memory effects*, Fossen (2011), and can be interpreted as filtered potential damping forces. These fluid memory effects are described by the convolution integral in the *Cummings equation* as

$$(\mathbf{M}_{RB} + \bar{\mathbf{A}})\ddot{\boldsymbol{\xi}} + \int_{\tau=-\infty}^t \bar{\mathbf{K}}(t - \tau)\dot{\boldsymbol{\xi}}d\tau + \bar{\mathbf{C}}\boldsymbol{\xi} = \boldsymbol{\tau}_{env} + \delta\boldsymbol{\tau}, \quad (3.50)$$

where $\bar{\mathbf{A}}$, $\bar{\mathbf{C}}$ and $\bar{\mathbf{K}}$ are defined in (Fossen, 2011, chap. 5.4), $\delta\boldsymbol{\tau}$ is the perturbed control input and $\boldsymbol{\xi}$ is defined in (Fossen, 2011, chap. 5.2). Note that (3.50) is defined in a seakeeping reference-frame and in the frequency domain. From Fossen (2011) the fluid memory effect in a BODY frame and time domain is defined as:

$$\boldsymbol{\mu} := \int_{\tau=0}^t \mathbf{K}(t - \tau)\delta\boldsymbol{\nu}d\tau. \quad (3.51)$$

3.6.3 Maneuvering Theory

When maneuvering theory is assumed (no wave forces) all the forces and moments can be approximated at one frequency of oscillation, such that the fluid memory effects can be neglected. By doing this the added mass and potential damping matrices will be approximated to a constant value. Now potential theory (see Section 3.6.1) can be used in hydrodynamic programs such as ShipX¹ to calculate the added mass and potential damping matrices.

¹<https://www.sintef.no/programvare/shipx/>

3.6.3.1 Added Mass and Coriolis

Fossen (2011) defined added mass as:

”... can be seen as a virtual mass added to a system because an accelerating or decelerating body must move some volume of the surrounding fluid as it moves through it. Moreover, the object and fluid cannot occupy the same physical space simultaneously.”

For surface vessels it is common to decouple the surge mode from the steering dynamics resulting in an added mass matrix, Fossen (2011),

$$\mathbf{M}_A = \mathbf{M}_A^\top = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & Y_{\dot{r}} & N_{\dot{r}} \end{bmatrix}. \quad (3.52)$$

As any motion in a stationary fluid creates motion in the fluid, Coriolis and centripetal forces are induced in the fluid when objects rotates in it, hence a centripetal force depending on the added mass is also introduced, Fossen (2011),

$$\mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} = \begin{bmatrix} Y_{\dot{v}}v_r r + Y_{\dot{r}}r^2 \\ -X_{\dot{u}}u_r r \\ (X_{\dot{u}} - Y_{\dot{v}})u_r v_r - Y_{\dot{r}}u_r r \end{bmatrix}. \quad (3.53)$$

Here the first term in the yaw moment is known as the *Munk moment*, and is known to have destabilizing effects.

3.6.3.2 Damping

In general the damping forces that act on a vessel contribute to both a linear and a nonlinear term, and it is convenient to write the damping as a sum of these two terms,

$$\mathbf{D}(\boldsymbol{\nu}_r) = \mathbf{D} + \mathbf{D}_n(\boldsymbol{\nu}_r), \quad (3.54)$$

where \mathbf{D} is the linear damping from potential damping and skin friction, and $\mathbf{D}_n(\boldsymbol{\nu}_r)$ is the nonlinear damping from quadratic drag.

From Fossen (2011) the linear damping matrix \mathbf{D} can be written as

$$\mathbf{D} = - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix}, \quad (3.55)$$

where the constants in the matrix are computed using hydrodynamic programs. The nonlinear terms of the damping are computed for each DOF, and the nonlinear surge damping is modeled by

$$X = -\frac{1}{2}\rho_{water}C_xA_xu_r|u_r|. \quad (3.56)$$

Here C_x is a current coefficient, A_x is the frontal projected area and u_r is the relative surge velocity. By utilizing the cross-flow drag principle (Faltinsen, 1990, pp. 187-197), the nonlinear force and moment in sway and yaw can be calculated by a summation of the 2D drag across the vessel according to

$$Y = -\frac{1}{2}\rho_{water}C_y^{2D}\frac{A_y}{L_{pp}}\int_{-\frac{L_{pp}}{2}}^{\frac{L_{pp}}{2}}(v_r + xr)|v_r + xr|dx, \quad (3.57)$$

$$N = -\frac{1}{2}\rho_{water}C_y^{2D}\frac{A_y}{L_{pp}}\int_{-\frac{L_{pp}}{2}}^{\frac{L_{pp}}{2}}x(v_r + xr)|v_r + xr|dx, \quad (3.58)$$

where C_y^{2D} is the 2D drag coefficient, A_y is the transversal area and $\frac{A_y}{L_{pp}}$ represents the vessel draft.

3.7 Summary of Vessel Dynamics

Putting together the results from Sections 3.3-3.6 a unified model of the dynamics can be summarized as

$$\begin{aligned} \dot{\eta} &= \mathbf{J}_{\Theta}(\eta)\boldsymbol{\nu}, \\ \mathbf{M}_{RB}\dot{\boldsymbol{\nu}}_r + \mathbf{M}_A\dot{\boldsymbol{\nu}}_r + \mathbf{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \\ &\quad \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{\mu} + \mathbf{G}\eta = \boldsymbol{\tau}_{env} + \boldsymbol{\tau}_{thr}, \end{aligned} \quad (3.59)$$

where $\boldsymbol{\nu}_r$ is the relative velocity defined as $\boldsymbol{\nu} - \boldsymbol{\nu}_c$, $\boldsymbol{\nu}_c$ is the current velocity and $\boldsymbol{\tau}_{env} + \boldsymbol{\tau}_{thr} = \boldsymbol{\tau}$.

THIS PAGE INTENTIONALLY LEFT BLANK

Motion Control Systems

This chapter will briefly present some concepts and theory from Guidance Navigation and Control (GNC) systems including the structure of a marine motion control system and briefly the implementation of these in modern computers. The second part of this chapter introduces in detail the different parts of a marine motion control system, this includes a controller, thruster allocation, filter, reference model and an observer.

4.1 Guidance Navigation and Control

In order to dynamically position a marine vessel, some form of a motion control system is needed. Fossen (2011) described the different classifications of motion control as:

- *Setpoint regulation (Point stabilization)*: A special case where the desired position and attitude are chosen to be constant.
- *Trajectory tracking*: The objective is to force the output $\mathbf{y}(t)$ to track a desired output $\mathbf{y}_d(t)$. This is most common for over-actuated vessels.
- *Path following*: To follow a predefined path independent of time. This is most common for under-actuated vessels.

A motion control system is usually constructed as three independent blocks denoted as the GNC systems, these system interacts with each other as illustrated in Figure 4.1. The main task of the (vessel motion) control system is to make the vessel follow a desired trajectory coming from the guidance control, this desired trajectory is defined in terms of the vessel's position, velocity and acceleration, Sørensen (2012), and is normally coming from the reference model. The navigation system is responsible for determining the vessel's position, which is usually performed with a Global Navigation Satellite

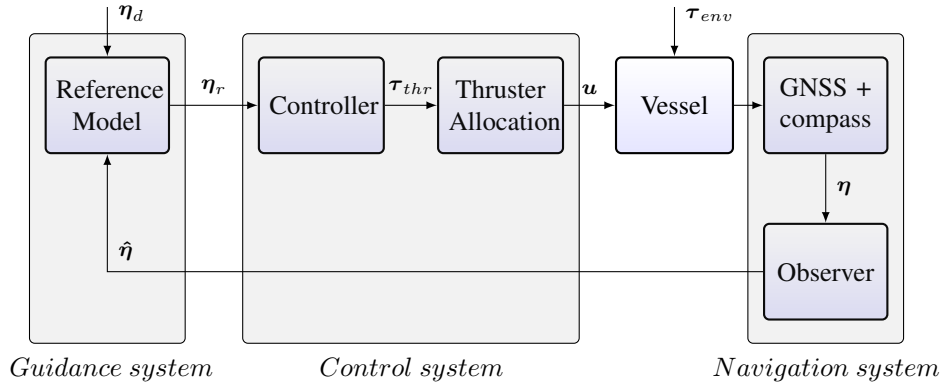


Figure 4.1: Illustration of a simplified GNC signal flow.

System (GNSS) system, combined with accelerometers/gyros. Fossen (2011) stated that an *autopilot* is a GNC system in its simplest form, it contains a guidance system (reference model), a navigation system (observer) and an autopilot (control system).

This thesis mainly focuses on DP, which according to Fossen (2011) and Sørensen (2012) also includes low-speed maneuvering ($|V| < 2 \text{ m/s}$). DNV (2011) stated that a DP vessel is:

”a vessel which automatically maintains its position and heading (fixed location or predetermined track) exclusively by means of thruster force”.

4.1.1 Computer-Based Control Systems

Sørensen (2012) suggested that marine control systems could be divided into two main areas: *real-time control and monitoring*, and *operational and business enterprise management*, see Figure 4.2. Here the *office systems* is the mission planning and/or supervision of the mission, in short, the tasks that are not time sensitive and computationally too heavy to run at real-time falls into this category. The *real-time control* category is as shown in Figure 4.2 divided into different time domains depending on their update frequency. Actuator control is typically local thrust- and propeller-control, plant control is typically the observer, controller and thruster allocation of the vessel, and local optimization is e.g. optimizing the DP setpoint according to the weather.

All of this is in today’s GNC systems implemented to run on computers, and they operate on continuous time at discrete time sampling instants, Sørensen (2012). This can as mentioned in Sørensen (2012) *”introduce deterioration’s of the control objective”* if the discrete time process is not accounted for, it is also worth mentioning that computer controlled systems opens up possibilities that were not possible in analog systems,

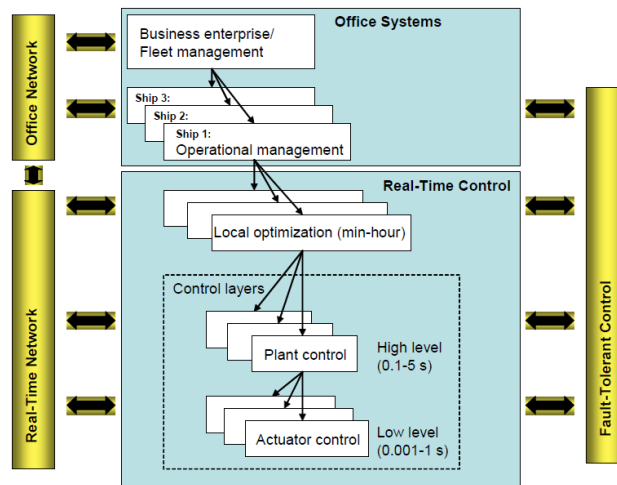


Figure 4.2: Illustration of a general marine control structure. Courtesy of Sørensen (2012).

Sørensen (2012). Since the computer and the process plant being controlled operate on different time instances as shown in Figure 4.3 the need for converters between discrete and continuous time is needed. The Analog to Digital (A/D) converter is a sampling function that samples the analog signal at given time intervals, the Digital to Analog (D/A) converter is a hold function, i.e. a zero-order hold, first-order hold etc.

A big concern when sampling, A/D converter, is at what sampling rate the discrete signal needs to be sampled, in order to reconstruct the original sampled signal. It is stated by the *Nyquist sampling theorem* that it is necessarily to sample at least twice per period, $T < \pi/\omega_M$, Shannon (1949). Note that if the signal has components that have a frequency higher than the Nyquist frequency a phenomenon called aliasing is introduced. Aliasing is when new frequency components are introduced from the sampling process, and these new frequencies can be described by

$$\omega_{sampling} = n\omega_s \pm \omega_M. \quad (4.1)$$

Here n is an arbitrary integer, ω_s is the sampling frequency and ω_M is the signal frequency, Sørensen (2012). To avoid aliasing an anti-aliasing filter should be introduced, the anti-aliasing filter is introduced by filtering out any frequencies above the Nyquist frequency from the signal.

Most of the sensor signals in today's control systems are provided by digital signals, analog signals will be converted before going to the computer for what is called signal processing. This is checking the signals for errors, or if there are more than one signal

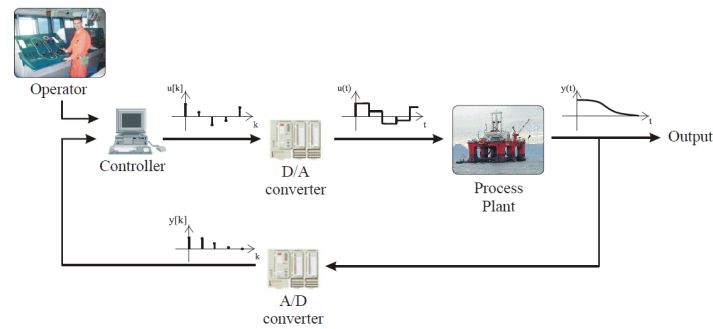


Figure 4.3: Illustration showing digital control of an analog system. Courtesy of Sørensen (2012).

is available voting and weighting of the signals should be implemented.

4.2 Control System

For a vessel the control system mainly consists of three different parts to generate the appropriate actuator/thruster demand to keep the vessel in the desired position:

- *Controller*: Generates force settings, i.e. desired control action.
- *Thruster Allocation*: Translates the desired control forces into actuator commands, e.g. Revolutions per minute (RPM), power, torque, azimuth angle.
- *Wave filter (observer)*: Recovers slowly-varying motion signals from the measurements, i.e. filter out the wave motion.

The most general and widely used controller in today's DP operations is a simple PID controller. The controller is setup using three decoupled controllers each controlling its own motion i.e surge, sway and yaw.

In the beginning of the 1960s a Single Input Single Output (SISO) PID controller was used to control the horizontal modes of motion. Later more advanced control theory was proposed and a Multiple Input Multiple Output (MIMO) controller was introduced with the introduction of the Kalman filter, Sørensen (2012). Nowadays there exists a number of variations to these controllers, and there exists both linear and nonlinear controllers for DP, they can be modified to obtain various degrees of stability, maximum rise time, minimal overshoot, etc. The controllers can also be modified with different FF- and FB-terms. These can be nonlinear if it is desirable to model nonlinear effects.

A MIMO linear PID controller in its simplest form can mathematically be described as

$$\tau_{pid} = \mathbf{K}_p \tilde{\boldsymbol{\eta}}(t) + \mathbf{K}_i \int_{\tau=0}^t \tilde{\boldsymbol{\eta}}(\tau) \delta\tau + \mathbf{K}_d \frac{\partial \tilde{\boldsymbol{\eta}}(t)}{\partial t}, \quad (4.2)$$

where $\mathbf{K}_{\{p,i,d\}} > 0$ are the controller gain matrices for the proportional, integral and derivative parts respectively. The error state $\tilde{\boldsymbol{\eta}}(t)$ is defined as the error between the desired state and the current state i.e. $\tilde{\boldsymbol{\eta}}(t) = \boldsymbol{\eta}_d(t) - \boldsymbol{\eta}(t)$. State-of-the-art linear PID controllers are common in today's DP systems, and (Fossen, 2011, chap. 12) quite extensively describes modern PID controllers. More advanced controllers such as back-stepping control, FB linearization and sliding mode control are described in (Fossen, 2011, chap. 13).

4.2.1 Acceleration Feedback

One way to improve the sensitivity to external disturbances is by adding an acceleration FB term. Consider a simple *mass-damper-spring* system,

$$m\ddot{x} + d\dot{x} + kx = \tau + w, \quad (4.3)$$

where the control law $\tau = \tau_{PID} - K_m \ddot{x}$ is given by a standard PID controller and an acceleration FB term $K_m > 0$. This gives

$$(m + K_m)\ddot{x} + d\dot{x} + kx = \tau_{PID} + w, \quad (4.4)$$

where the effect of the acceleration FB term is an *increased* mass from m to $(m + K_m)$. In addition by dividing through by the new mass the acceleration FB also reduces the gain of the disturbance w from $1/m$ to $1/(m + K_m)$. This will as stated above improve the controllers performance towards external disturbances.

4.2.2 Tuning of the Controller

Tuning of the controller is essential for an optimal response as mentioned above. The system information available determines which method can be used to tune the controller. For instance if a model of the system is unavailable, but the physical system is available, manual tuning methods like the the Ziegler-Nichols method, Ziegler and Nichols (1993), can be applied. The main drawback with manual tuning methods is that they are time consuming and it can be difficult to get the parameters optimally tuned. On the other hand if the system model is known, tuning the controller gains with help of *pole-placement* can be a good option. Fossen (2011) suggested a pole-placing algorithm

based on the controller design parameters; bandwidth ω_b and relative damping ζ , this can be seen in Algorithm 1. Fossen (2011) also stated that the relative damping ratio ζ for marine vessels are typically in the range 0.8 to 1, and that the controller bandwidth typically are 0.01 rad/s for large tankers and 0.1 rad/s for smaller vessels.

Algorithm 1 PID pole-placement algorithm.

- 1: Specify the bandwidth $\omega_b > 0$ and the relative damping ratio $\zeta > 0$.
 - 2: Compute the natural frequency: $\omega_n = \frac{1}{\sqrt{1-2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}}\omega_b$.
 - 3: Compute: $K_p = (m + K_m)\omega_n^2$.
 - 4: Compute: $K_d = 2\zeta\omega_n(m + K_m)$.
 - 5: Compute: $K_i = \frac{\omega_n}{10}K_p$.
-

4.3 Thruster Allocation

For a marine vessel it is necessary to distribute the generalized forces $\boldsymbol{\tau} \in \mathbb{R}^n$ coming from the controller to the actuators on the vessel in form of a control input $\boldsymbol{u} \in \mathbb{R}^r$. This is accomplished by the use of a thruster allocation. The system is called *over-actuated* when $r > n$, and *under-actuated* when $r < n$, Fossen (2011).

The thruster forces in BODY provided from each propeller on the vessel can be assumed as

$$\boldsymbol{f}_{thr}^b = \boldsymbol{K}\boldsymbol{u}, \quad (4.5)$$

where \boldsymbol{u} is a vector of control inputs, usually consisting of propeller pitch and RPM, and \boldsymbol{K} is a force coefficient matrix relating the propeller input to the actual force provided. By using a thrust configuration matrix \boldsymbol{T} , which is determined based on the geometrical properties of each propeller, i.e. where on the vessel they are located and in which direction they provide thrust, the actuator forces can then be calculated as

$$\boldsymbol{\tau}_{thr} = \boldsymbol{T}\boldsymbol{K}\boldsymbol{u}. \quad (4.6)$$

If there are rudders and/or azimuth thruster the thrust configuration matrix will depend on the angles $\boldsymbol{\alpha}$, i.e. $\boldsymbol{T} = \boldsymbol{T}(\boldsymbol{\alpha})$. If there exists a number of thrusters in each DOF, a weighting matrix \boldsymbol{W} can be added to penalize the use of some thrusters compared to others. From Fossen (2011) it can be shown that the control input can be calculated as

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{T}_w^\dagger\boldsymbol{\tau}_{thr}, \quad (4.7)$$

where $\boldsymbol{T}_w^\dagger = \boldsymbol{W}^{-1}\boldsymbol{T}^\top(\boldsymbol{T}\boldsymbol{W}^{-1}\boldsymbol{T}^\top)^{-1}$ is the pseudo-inverse. If the configuration matrix

T is square or non-square with $r > n$, it is possible to find an *optimal* distribution of the forces f_{thr}^b for each DOF. This can for instance be performed with a Least-Squares (LS) optimization problem, see Fossen and Sagatun (1991).

4.4 Filter

The disturbances affecting a vessel can be separated into two different components; Low Frequency (LF) and Wave Frequency (WF). In general the LF motions are assumed to be caused by second-order mean and slowly varying wave loads, current loads, wind loads and thruster forces. WF motions are assumed to be caused by first-order wave loads. The WF have zero mean force while the LF motion can have a mean force different from zero. This mean force will result in drift forces which needs to be counteracted by the controller. Filtering out the WF is beneficial since the controller do not need to correct for each incoming wave and only control the LF motion, this will in turn reduce the wear and tear of the vessels propulsion system, Fossen (2011). Since a filter already suppresses high frequencies it can with great benefits also be used to suppress noise in the system measurements.

There are different types of filters, the most used to filter out WF is a Low Pass (LP) filter. A first-order LP filter with time constant T_f will suppress frequencies above $1/T_f$, Fossen (2011), and will have the transfer function

$$h_{lp}(s) = \frac{1}{T_f s + 1}. \quad (4.8)$$

When using a LP filter it is important to have good knowledge of the WF and the control system frequencies. If the control system frequencies are close to $1/T_f$, the control system frequencies will be suppressed, this is particularly a problem for smaller vessels. One possible solution to this problem is to use a LP filter in cascade with a notch filter, Fossen (2011). (4.9) shows the transfer function for a second-order notch filter, this notch filter will suppress frequencies at ω_{notch} . The notch frequency should be chosen equal to the peak frequency ω_0 of the spectrum for the marine vessel at zero speed. Additionally the design parameter $0 < \zeta < 1$ is used to control the magnitude of the notch filter. Note that additional phase lag is introduced when using a notch filter in cascade with another filter.

$$h_n(s) = \frac{s^2 + 2\zeta\omega_{notch}s + \omega_{notch}^2}{(s + \omega_{notch})^2}. \quad (4.9)$$

4.5 Reference Model

A reference model calculates trajectories for the desired vessel motion for each DOF, and is essential in order to be able to perform controlled movements and rotations of the vessel. The trajectory should take into account the model dynamics to ensure the system is capable of doing the maneuver. As a result of using a reference model the propulsion system usage will be smoother hence reduce the wear and tear of the propulsion system and make the motions of the vessel more damped. When modelling a reference model, it is important that the bandwidth of the reference model is chosen lower than the bandwidth of the motion control system, i.e the controller is faster than the reference model. This is needed to obtain satisfactory tracking performance and stability. It can also be advantageous to saturate the velocity and/or acceleration, depending on the operation type; trajectory tracking, fixed point DP etc.

The simplest form of a reference model is obtained by using a LP filter, where it is important to implement the velocity and acceleration limitations. In order to increase performance higher-order LP filter(s) can be used, a LP filter of second-order can be formulated as

$$h_{lp}(s) = \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2}. \quad (4.10)$$

(Sørensen, 2012, chap. 8.3.2) described a simple *third-order* linear reference model which calculates the desired trajectory for acceleration, velocity and position. The drawback to a linear reference model is that the time constant will be the same for any distance to a new setpoint. Another way to implement a reference model is described in Fernandes (2015). Here the reference model calculates an optimal velocity and acceleration based on the length and angle between the vessel and the desired setpoint. This model splits into four phases where the acceleration and deceleration occur in different phases, these will be optimal and saturated for reaching the setpoint in the desired amount of time. Fossen (2011) described a simple mass-damper-spring system with saturating elements to calculate a third-order reference model. Fossen (2011) discussed that a drawback to a linear reference model is that the time constants will give a satisfactory response for one input, but totally changes the behaviour with other inputs (larger or smaller amplitudes in the input). One solution to this problem is to use *amplitude gain scheduling* to change the reference model parameters according to the input.

4.6 Observer

An observer or state estimator produces the state of a system from measurements of inputs and outputs, Sørensen (2012). The observers purpose is to reconstruct states that is either impossible or to expensive to measure, and in marine vessels the observer should estimate velocities, current and wave drift forces. The simplest form of an observer is called a fixed gain observer, e.g. a Luenberger observer. Consider a linear time-invariant system, then an observer can be constructed by mimicking the system dynamics,

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}} &= \mathbf{A}\hat{\boldsymbol{x}} + \mathbf{B}\boldsymbol{u} + \boldsymbol{\gamma}(\boldsymbol{y}, \hat{\boldsymbol{y}}), \\ \hat{\boldsymbol{y}} &= \mathbf{H}\hat{\boldsymbol{x}},\end{aligned}\tag{4.11}$$

where $\hat{\cdot}$ represents the estimate of the state \cdot and the injection term $\boldsymbol{\gamma}$ is chosen to make $\hat{\boldsymbol{x}} \rightarrow \boldsymbol{x}$ as $t \rightarrow \infty$. If the injection term $\boldsymbol{\gamma}$ is chosen as

$$\boldsymbol{\gamma}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \mathbf{K}(\boldsymbol{y} - \hat{\boldsymbol{y}}),\tag{4.12}$$

when \mathbf{K} is a constant matrix of observer gains, the resulting observer is a Luenberger observer. However if an observer is to succeed in estimating the states the system needs to be *observable*. This can be understood as if it is possible to reconstruct the state \boldsymbol{x} by knowing all the external inputs \boldsymbol{y} and \boldsymbol{u} . Mathematically it is defined according to an observability matrix \mathcal{O} to be of full column rank,

$$\mathcal{O} := [\mathbf{H}^\top \mid \mathbf{A}^\top \mathbf{H}^\top \mid \cdots \mid (\mathbf{A}^\top)^{n-1} \mathbf{H}^\top].\tag{4.13}$$

Asymptotic convergence of $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$ can be proven (see Fossen (2011)) for a Luenberger observer. Because of the high number of states and increased demand in tracking performance, more advanced observers are often used on marine vessels, e.g. Kalman filter or nonlinear observer.

4.6.1 Nonlinear Passive Observer

The major drawback of a Kalman filter is that it is difficult and time-consuming to tune. The main reason being the many different co-variance tuning parameters that can be difficult to relate to the physical quantities of the model, Fossen (2011). The number of tuning parameters can be drastically reduced by using a nonlinear passive observer. Another great advantage of using a nonlinear passive observer is that there is no need to linearize the equations of motion around a predefined set of yaw angles (typically 36 in increments of 10°), as is needed if linear (Kalman filter) theory is used, also Global

Exponential Stability (GES) cannot be guaranteed when linear theory is used.

A nonlinear passive observer will guarantee GES of all the estimated states. Hence only one set of observer gains is needed to cover the whole state-space, and only one set of parameters requires tuning. Passivity implies that the phase of the error dynamics is limited by 90° , if this is satisfied it will result in excellent stability properties. Fossen and Strand (1999) proposed a nonlinear passive observer as

$$\dot{\hat{\boldsymbol{\xi}}} = \mathbf{A}_w \hat{\boldsymbol{\xi}} + \mathbf{K}_1(\boldsymbol{\omega}_0) \tilde{\mathbf{y}}, \quad (4.14a)$$

$$\dot{\hat{\boldsymbol{\eta}}} = \mathbf{R}(\psi) \hat{\boldsymbol{\nu}} + \mathbf{K}_2 \tilde{\mathbf{y}}, \quad (4.14b)$$

$$\dot{\hat{\mathbf{b}}} = -\mathbf{T}^{-1} \hat{\mathbf{b}} + \mathbf{K}_3 \tilde{\mathbf{y}}, \quad (4.14c)$$

$$\mathbf{M} \dot{\hat{\boldsymbol{\nu}}} = -\mathbf{D} \hat{\boldsymbol{\nu}} + \mathbf{R}^\top(\psi) \hat{\mathbf{b}} + \boldsymbol{\tau}_{thr} + \boldsymbol{\tau}_{env} + \mathbf{R}^\top(\psi) \mathbf{K}_4 \tilde{\mathbf{y}}, \quad (4.14d)$$

$$\hat{\mathbf{y}} = \hat{\boldsymbol{\eta}} + \mathbf{C}_w \hat{\boldsymbol{\xi}}. \quad (4.14e)$$

Here $\hat{\boldsymbol{\xi}}$ is a first-order wave estimation and $\hat{\mathbf{b}}$ is a bias estimator that is used to model any uncertainties in the model, e.g. drift-, wind- and current-forces. $\tilde{\mathbf{y}} = \mathbf{y} - \hat{\mathbf{y}}$ are the estimation error between the estimate $\hat{\mathbf{y}}$ and the measured state \mathbf{y} . In order to prove Uniform Global Exponential Stability (UGES), $\mathbf{T} > 0$ is needed in (4.14c). This is because when $\mathbf{T} > 0$ the model also does LP filtering of the bias instead of only integrating the white noise term $\mathbf{K}_3 \tilde{\mathbf{y}}$ as is done when $\mathbf{T} = 0$, and this can be proven to only give asymptotic stability, Fossen (2011). Note that $\mathbf{K}_1(\boldsymbol{\omega}_0)$ is a function of the wave spectra peak frequency $\boldsymbol{\omega}_0 = [\omega_{01}, \omega_{02}, \omega_{03}]$ in surge, sway and yaw.

When deriving the passivity of this system (4.14a), (4.14b) and (4.14e) is written in state-space form, according to Fossen (2011),

$$\begin{aligned} \dot{\hat{\boldsymbol{\eta}}}_0 &= \mathbf{A}_0 \hat{\boldsymbol{\eta}}_0 + \mathbf{B}_0 \mathbf{R}(\psi) \hat{\boldsymbol{\nu}} + \mathbf{K}_0(\boldsymbol{\omega}_0) \tilde{\mathbf{y}}, \\ \hat{\mathbf{y}} &= \mathbf{C}_0 \hat{\boldsymbol{\eta}}_0, \end{aligned} \quad (4.15)$$

where $\hat{\boldsymbol{\eta}}_0 = [\hat{\boldsymbol{\xi}}^\top, \hat{\boldsymbol{\eta}}^\top]^\top$, $\mathbf{K}_0(\boldsymbol{\omega}_0) = \begin{bmatrix} \mathbf{K}_1(\boldsymbol{\omega}_0) \\ \mathbf{K}_2 \end{bmatrix}$, $\mathbf{B}_0 = \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ \mathbf{I}_{3 \times 3} \end{bmatrix}$ and $\mathbf{C}_0 = [\mathbf{C}_w, \mathbf{I}_{3 \times 3}]$. If the estimation errors are defined as $\tilde{\boldsymbol{\nu}} = \boldsymbol{\nu} - \hat{\boldsymbol{\nu}}$, $\tilde{\mathbf{b}} = \mathbf{b} - \hat{\mathbf{b}}$ and $\tilde{\boldsymbol{\eta}}_0 = \boldsymbol{\eta}_0 - \hat{\boldsymbol{\eta}}_0$ the error dynamics becomes

$$\dot{\tilde{\boldsymbol{\eta}}}_0 = [\mathbf{A}_0 - \mathbf{K}_0(\boldsymbol{\omega}_0) \mathbf{C}_0] \tilde{\boldsymbol{\eta}}_0 + \mathbf{B}_0 \mathbf{R}(\psi) \tilde{\boldsymbol{\nu}}, \quad (4.16a)$$

$$\dot{\tilde{\mathbf{b}}} = -\mathbf{T}^{-1} \tilde{\mathbf{b}} - \mathbf{K}_3 \tilde{\mathbf{y}}, \quad (4.16b)$$

$$\mathbf{M} \dot{\tilde{\boldsymbol{\nu}}} = -\mathbf{D} \tilde{\boldsymbol{\nu}} + \mathbf{R}^\top(\psi) \tilde{\mathbf{b}} - \mathbf{R}^\top(\psi) \mathbf{K}_4 \tilde{\mathbf{y}}. \quad (4.16c)$$

The dynamics in (4.16c) is rewritten as

$$\mathbf{M}\dot{\tilde{\nu}} = -\mathbf{D}\tilde{\nu} - \mathbf{R}^\top(\psi)\tilde{z}, \quad (4.17)$$

where $\tilde{z} = \mathbf{K}_4\tilde{\mathbf{y}} - \tilde{\mathbf{b}}$. By defining a new state vector the system matrices (4.16) can be written in a compact form

$$\begin{aligned} \dot{\tilde{\mathbf{x}}} &= \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{R}(\psi)\tilde{\nu}, \\ \tilde{z} &= \mathbf{C}\tilde{\mathbf{x}}, \end{aligned} \quad (4.18)$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 - \mathbf{K}_0(\omega)_0\mathbf{C}_0 & \mathbf{0}_{9 \times 9} \\ -\mathbf{K}_3\mathbf{C}_0 & -\mathbf{T}^{-1} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{0}_{3 \times 3} \end{bmatrix}$, $\mathbf{C} = [\mathbf{K}_4\mathbf{C}_0, -\mathbf{I}_{3 \times 3}]$ and $\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\eta}_0 \\ \tilde{\mathbf{b}} \end{bmatrix}$.

Now the observer system can be viewed as two linear and separate blocks, Fossen and Strand (1999),

$$\mathcal{H}_1 : \left\{ \mathbf{M}\dot{\tilde{\nu}} = -\mathbf{D}\tilde{\nu} - \mathbf{R}^\top(\psi)\tilde{z} \right. , \quad (4.19)$$

$$\mathcal{H}_2 : \left\{ \begin{aligned} \dot{\tilde{\mathbf{x}}} &= \mathbf{A}\tilde{\mathbf{x}} + \mathbf{B}\mathbf{R}(\psi)\tilde{\nu} \\ \tilde{z} &= \mathbf{C}\tilde{\mathbf{x}} \end{aligned} \right. . \quad (4.20)$$

The observer system can now be proven to be passive if one block is strictly passive and the other is passive, Fossen (2011) proved that \mathcal{H}_1 is strictly passive by using Lyapunov theory and \mathcal{H}_2 is passive by the *Kalman-Yakubovic-Popov* lemma.

The observer gain matrices $\mathbf{K}_1 \in \mathbb{R}^{6 \times 3}$, $\mathbf{K}_{2,3,4} \in \mathbb{R}^{3 \times 3}$ are tuned to satisfy the *Kalman-Yakubovic-Popov* lemma in order to obtain the needed stability (passivity and GES). In Fossen (2011) it is mentioned that block \mathcal{H}_2 describes three decoupled systems in surge, sway and yaw indicating that the gain matrices should have a diagonal structure,

$$\mathbf{K}_1(\omega_0) = \begin{bmatrix} \text{diag}\{K_{11}(\omega_{01}), K_{12}(\omega_{02}), K_{13}(\omega_{03})\} \\ \text{diag}\{K_{14}(\omega_{01}), K_{15}(\omega_{02}), K_{16}(\omega_{03})\} \end{bmatrix}, \quad (4.21a)$$

$$\mathbf{K}_2 = \text{diag}\{K_{21}, K_{22}, K_{23}\}, \quad (4.21b)$$

$$\mathbf{K}_3 = \text{diag}\{K_{31}, K_{32}, K_{33}\}, \quad (4.21c)$$

$$\mathbf{K}_4 = \text{diag}\{K_{41}, K_{42}, K_{43}\}. \quad (4.21d)$$

The gains are tuned to achieve a notch-filtering effect of the peak wave frequency re-

sulting in the following formula for the gains $\mathbf{K}_1(\omega_0)$ and \mathbf{K}_2 ,

$$\mathbf{K}_{1i}(\omega_{0i}) = -2(\zeta_{ni} - \lambda_i) \frac{\omega_{ci}}{\omega_{0i}}, \quad (4.22a)$$

$$\mathbf{K}_{1(i+3)}(\omega_{0i}) = -2\omega_{0i}(\zeta_{ni} - \lambda_i), \quad (4.22b)$$

$$\mathbf{K}_{2i} = \omega_{ci}. \quad (4.22c)$$

Here $\zeta_{ni} > \lambda_i$ determines the notch frequency, and $\omega_{ci} > \omega_{0i}$ determines the filter cutoff frequency (typical values: $\zeta_{ni} = 1.0$ and $\lambda_i = 0.1$). The *Kalman-Yakubovic-Popov* lemma is also used to achieve a guarantee of less than 90° , from Fossen (2011) the following rule is proposed for tuning T_i , K_{3i} and K_{4i} ,

$$1/T_i \ll \mathbf{K}_{3i}/\mathbf{K}_{4i} < \omega_{0i} < \omega_{ci}. \quad (4.23)$$

Improvements of Simulator of Arctic Marine Structures

Several improvements and additional features have been added to the SAMS source code during this thesis, and this chapter will go through the most important steps in how the simulation environment has been modified, this will include:

- Modification of the SAMS source code to facilitate multiple free moving vessels.
- Testing and verification of both the SAMS source code and the TCP interface.
- Performance enhancements in SAMS.
- Modification of input to the simulator and post processing of data.

Most of the added features were implemented in accordance with the predefined scope and inputs from the co-advisor Jon Bjørnø, however some features like the performance enhancements were implemented as a consequence of problems that was discovered during testing and verification of the code.

5.1 Simulator of Arctic Marine Structures

SAMS is a numerical simulator originally designed as a towing tank simulator to simulate structures and vessels being towed in an ice field and in turn get the ice loads on the respective structures/vessels. The simulator is an adaptive simulation environment by taking a configuration file *SAMS.itconfig* as input, meaning that the ice field, structures and properties can be exchanged between simulations, an example of a configuration file can be seen in Appendix G.1. SAMS uses the parameters specified in the configu-

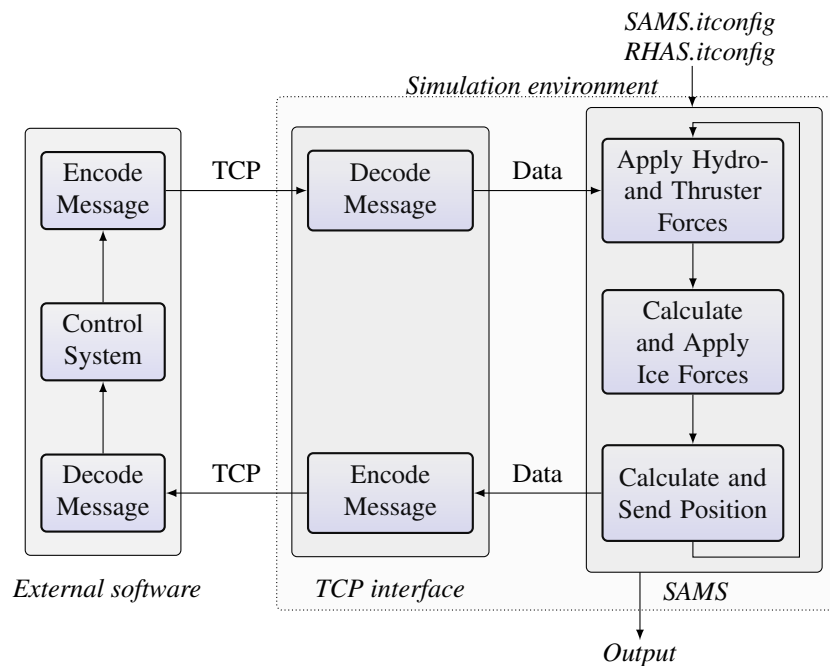


Figure 5.1: High level overview of the simulation environment together with the external software.

ration file to simulate a run through the towing tank writing selected states to an output file.

The SAMS code used in this thesis is a developer edition and has constantly been updated. The code is developed by Arctic Integrated Solutions AS (ArcISO), a spin of company from Norwegian University of Science and Technology (NTNU), where a first release of the simulator is planned for June of 2018. The simulator is built upon the foundation of a simulator named NIT developed by Sustainable Arctic Marine and Coastal Technology (SAMCoT) that is based on the work of Lubbad and Løset (2011), among others.

A high level flow chart of the simulation environment can be seen in Figure 5.1. Here it can be seen how SAMS has been extended with the library, *TCP interface*, that enables communication over TCP. The library will be further presented in Chapter 6 while the modifications to the SAMS code itself will be presented in Section 5.2. The external software is seen as a separate part and this will be presented in Chapter 7.

5.2 Modification of SAMS

The SAMS code is under constant revision and thus a modular and flexible code is needed. The idea behind the modifications has been to have it as a library that can easily be added or removed as necessary. Unfortunately this has not been fully possible as the SAMS code do not have any standard way of communication through a library. This section will thus be about the modifications implemented in the original SAMS code.

The integration and modification of code have been done in such a way that it is still possible to execute the original code, i.e. a towing tank simulator. The choice between a towing tank and a free moving vessel can easily be chosen from a separate configuration file called *RHAS.itconfig*, this configuration file is made for this thesis and can be seen in Appendix G.2. If by any reason the configuration file is not found the simulation returns to its default state as a towing carriage simulator. If however the configuration is set to not use the towing carriage the towing carriage is deactivated completely from the simulator and the TCP interface is activated.

5.2.1 Additional Vessels

The communication interface supports an unlimited¹ number of vessels each with up to 30 thrusters each. To make this possible a method for adding additional vessels was made and integrated into the SAMS code. The additional vessels are specified in the *RHAS.itconfig* file and can in this way have different properties and work independent of each other. Though additional vessels have been added and SAMS have been modified to handle them, contact between vessels are not considered and will result in a fatal error from SAMS.

In the initialization process the vessels might not be initialized in its full equilibrium position. This might be due to a difference in the way SAMS calculates the hydrostatic forces, numerical errors or inaccuracy in the model itself. This was solved by making a new parameter which calls an additional function in SAMS to calculate and re-initialize the structure at its equilibrium position, removing potentially unwanted oscillations.

5.2.2 Hydrodynamics

In addition to the modifications mentioned over it is also possible to use custom hydrodynamic forces for each vessel. This can be activated in the *RHAS.itconfig* file and will

¹TCP have a limitation of 65535 ports, some which might already be in use.

deactivate all hydrodynamic forces on all vessels calculated from the simulator. This means that the hydrodynamic forces needs to be calculated outside of the simulator and sent in for each vessel. Note that by using this, wind forces on the vessels are also deactivated as well as current and propeller wash from other vessels. If these forces are wanted the hydrodynamics either has to be calculated in SAMS or these forces needs to be added to the hydrodynamic forces sent into the simulator for each vessel.

Ice floes also experience hydrodynamic forces, but these are always calculated in SAMS. However this is often not enough to get good results in drift ice as the ice floes always starts with zero velocity. As of this a variable have been added to the *RHAS.itconfig* file to set an initial velocity of the ice field at the start of the simulation. This has proven to give more even results throughout the ice field as the velocity of the ice floes are more constant throughout the simulation.

5.2.3 Removal of Floes

In addition to modification and adding of features the performance has undergone some modifications. This was necessary as the simulation needed often compromises several hours, which can take up to several days to simulate. To reduce the simulation time removal of floes outside a specific scope was implemented. This is mainly designed for longer simulations with ice drift as the ice is of no interest when it has drifted past the vessels. In addition removal of fractured floes that has a lower mass than that specified by the user has been made available through the *SAMS.itconfig* file.

5.2.4 Simulator Output

The changes mentioned above prompted the need for a redesign of the output from the simulator since this is closely linked to the towing carriage and does not support multiple vessels/structures. Modifications were made such that each vessel has its own output file where parameters are stored for later use. Table E.1 shows an overview of the different parameters that are stored and their format. This include the force output for each thruster and hydrodynamic forces if the custom hydrodynamic mentioned above is activated.

Together with this a logging feature that logs all the floes that is removed and their size is implemented. The logging feature also outputs the location of the removed floes for the floes that are removed outside the scope and the simulation time step for the removal of each floe, Table E.2 shows an overview of the parameters logged and their format. This makes it possible to generate statistics to compare different IM approaches, as well

as look at the efficiency of ice-breaking. Post processing of the data to estimate ice forces from the removed floes can easily be accomplished.

5.2.5 Testing and Verification

Together with the changes and modifications mentioned above, a large part of the workload was on unit testing and verification of the code itself. This was done in an *Agile* fashion where new functionality was added and tested continuously. This made troubleshooting easier as only parts of the code was affected by each change. At the same time testing of new code provided by ArcISO to get enhanced performance and better results has been part of the integration of the code. This is especially true since the use of SAMS for this thesis is highly different than the intended use of SAMS, as such problems have arisen and a large part of the implementations in this thesis is a direct result of fixes for these problems.

Since the TCP interface and its integration into SAMS is intended to live beyond this thesis requirements have been put on coding style and documentation. The coding style and code documentation follows *Ceetron Solutions C++ Programming Style Guidelines, version 1.1* as close as possible as requested by ArcISO, though not openly available the guidelines are in large the same as can be found in most C++ programming books.

5.3 Ice Fields

The ice fields used in the simulator environment needs to be created and this involves using a program from SAMCoT and Ceetron Solutions AS named IceMaker. IceMaker creates ice fields by a power law and the method is described in for instance Yulmetov et al. (2016). For larger ice fields this has proven to be problematic due to the way IceMaker resolves overlapping floes. IceMaker will try to resolve any overlap by giving the floes small impulses, but when the number of floes increases this process gets harder, hence more time consuming. Thus a solution where the ice field are split into multiple parts and stitched together in post processing have been implemented.

5.3.1 Making of Ice Field Sections

Making smaller ice fields sections in IceMaker proved to be to difficult due to the need to put an offset on the position, which in IceMaker can only be achieved in x direction. Instead a program developed by Wenjun Lu, hereafter called the Weibull IceMaker, was modified to allow for offset of the ice fields. The Weibull IceMaker uses a Weibull distribution to make ice floes, additionally the program allows for controlling the roundness

of the floes. This gives a greater flexibility such that the floes can be made sharp (triangles, rectangles) like in IceMaker, more round or a mixture of the two, providing a more realistic ice field.

In addition to the offset modification, removal of floes with a radius under certain limit as well as output describing the ice field in the same way as normal ice observations has been implemented. The last modification makes verification of the ice field against observed data easy and efficient. After the ice field are created by the Weibull IceMaker it needs to go through IceMaker in order to resolve floe overlap and be created in the right format.

5.3.2 Stitching of Ice Fields

The ice field file can compromise tens of thousand of code lines, as of this manual stitching of the data is impractical. For this reason a JavaScript program was developed for this thesis to stitch the different ice fields together. The code for this can be seen in Appendix G.3 and takes the file path of the different ice fields and stitch it together into one single ice field. The ice field can then be used in SAMS as any other ice field. Though it with this method is possible to make ice fields that consist of a huge number of ice floes precaution should be taken, SAMS will have big problems with ice fields consisting of more than 10 000 floes, in addition the computational time in SAMS increases with an increase in the number ice floes, hence any simulation should try to avoid unnecessarily large ice fields.

5.4 Note on Copyright of SAMS

As SAMS is a commercial product the source code and the changes carried out in this thesis can not be shared or shown. The results obtained and the TCP interface itself can though be shown and is appended in the digital appendix. For access to SAMS either with or without the TCP interface one should contact ArcISo directly.

Implementation of a Transmission Control Protocol Interface

This chapter will go through the basic theory of TCP in addition to basic network and socket theory. The last part of the chapter will go through the implementation of the TCP interface which works as a connection point between SAMS and the external software.

6.1 TCP/IP

All information in this section is taken from Fall and Stevens (2012), unless otherwise stated.

In this section three important new concepts need to be understood; data, datagram and packet. Data is in its simplest form a 0 or 1 (binary data) that when put together can be used to represent complex information. Data can both be a single *bit* or multiple *bytes*, and can represent both information and *metadata* i.e. data about the data. As of such the other concepts above can both include data and be data themselves. Datagram is used for data on the data link layer, while packet is used for data in the TCP and Internet Protocol (IP) layer. The layers, TCP and IP will be presented in more details in the following paragraphs.

The Transmission Control Protocol/Internet Protocol (TCP/IP) is a protocol designed for reliable transmission over a network. The protocol is in reality two protocols combined together, namely TCP and IP. TCP is the protocol that handles reliable end-to-end communication, this includes handling of duplication, loss and reordering of data. The IP protocol on the other hand handles the transport of data from one place to another.

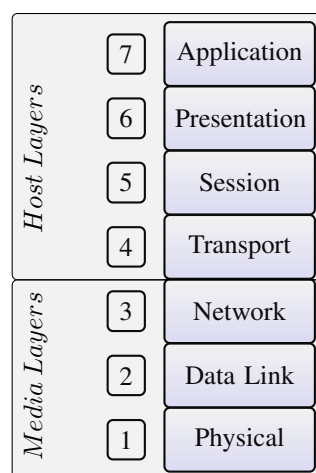


Figure 6.1: The Open Systems Interconnection model.

A network is divided into layers, and part of understanding TCP/IP lies in understanding the way a network is built up. The Open Systems Interconnection (OSI) model as shown in Figure 6.1 provides a simple model for understanding the layers of a standard network. It should be mentioned that TCP/IP is usually considered to only consist of four or five layers, nevertheless the functionality are in large the same. The layers of the OSI model are given a number from bottom to top, starting at one and ending at seven. All layers contains information for the layers above and for efficiency it is therefore an advantage to stay at the lowest layer possible. When data is sent it is usually sent from an application at layer seven, this data is then passed down the layers until it reaches the physical layer at level one which sends the data on the network. Before the sending can happen the data goes through layer four which consist of TCP and layer three which consist of IP.

The details of layer one and two will not be covered, but their purpose is to send IP-datagrams. A good analogy to IP is the postal system¹. Everyone gets its own IP-address on a network, and if one person (computer) decides to send a packet of data to another person (computer) it will write the address of the recipient and put it in a mailbox. From there the sender knows nothing about the route the packet will take or even if it will get to the destination. IP works in almost exactly this way, it packages the data into one packet and send it to the data link layer. Here the package might be broken up into smaller pieces and maybe sent in different paths until it might get to the receiver at the other end. When on the other end the packets might arrive in a different order than when they were sent and the data in the packages might not be the same as when they left, i.e. the data might have been corrupted on the way.

¹The post office analogy: <http://bpastudio.csudh.edu/fac/lpress/471/hout/netech/postofficelayers.htm>

The problems with the best effort delivery system of IP described above, is fixed with TCP which is part of the fourth layer in the OSI model. As TCP provides reliable transfer it needs to verify that the data is actually received by the recipient and that the data is intact. To do this TCP relies on ACKs and a checksum. An ACK is a small packet that is sent back from the recipient to confirm that the package has been received correctly, while a checksum is a function that derives a number from the data to detect errors. When a package is received a checksum is derived from the received packet and if this does not match the checksum sent with the data, then the data is corrupt and an ACK is not sent back to the receiver. If the sender then does not receive an ACK within a given time it will re-transmit the packet. Problems can arise when the sender does not wait long enough for an answer or the ACK is lost on the way back to the sender. To account for this every packet gets a unique sequence number, this provides the receiver with a way of determining if a specific packet has been received before and the sender with a possibility to verify which packet the ACK is for.

6.1.1 Socket

A socket is an endpoint in a communication network for sending and receiving data. A socket can also be used as a term for the combination of an IP-address and a port number, Fall and Stevens (2012), but in this thesis this will be referred to as socket address. Sockets is what allows for communication between different processes, either on a local computer or on a network. The use of sockets are done using an Application Programming Interface (API)², more specifically a socket API. Socket APIs allows for communication between applications using the TCP/IP communication protocol and is the network standard for TCP/IP communication, IBM (2013c). There exists several different socket APIs, the most known and de facto standard for sockets today is the Berkeley socket API, Fall and Stevens (2012). The socket API is located between the application layer on level seven and the transport layer on level four in the OSI model in Figure 6.1. The socket API is not a distinct layer in the model, but socket APIs are what allows different applications to interact with the transport and network layers, IBM (2013a).

Socket APIs are commonly used in a client-server interaction with the most typical being a server on one computer and clients on several different computers. On the server side the socket API establishes a socket, binds the socket to an IP-address and a port and waits (listens) for any clients to connect to the socket using the known address and port (socket address). When the client connects, a socket pair will be established and a data exchange can begin, IBM (2013a). It is worth mentioning that the client

²<https://www.mulesoft.com/resources/api/what-is-an-api>

and server can use different socket APIs but the socket type must be the same, some examples of different sockets from IBM (2013b) are:

- *Stream Sockets*: Also known as connection-oriented sockets and are used with the TCP protocol, this enables the transmission of data to be reliable and in order.
- *Datagram Sockets*: Also known as connection-less sockets and are used with User Datagram Protocol (UDP), each packet of data is individually addressed and sent, hence order and reliability is not guaranteed.
- *Raw Sockets*: Allow for direct sending of data by bypassing the transport layer in Figure 6.1. They are used in internet security, however they have been proven to be bug ridden and limited in use, faqs.org (2014).

6.2 TCP Interface

The TCP interface is implemented as a stand-alone library for modularity and easy maintenance, this reduces the TCP's interface dependency on SAMS while still allowing SAMS to utilize its functions. It also opens the possibility for the library to be used with other software with few to no modifications. The interface utilizes the Winsock 2³ socket API to set up and maintain a server that the external software clients can connect to. A high level description of the TCP interface and its function calls can be seen in Figure 6.2.

6.2.1 Simulation Loop

During the initialization of the simulation a *setup* function is called with the port number and the number of vessels specified in the *RHAS.itconfig* file. The setup function listens on the specified ports for a client in the external software to connect, this is repeated until all vessels are connected and the simulation loop is then started.

The main simulation loop in SAMS calculates position, velocity and ice forces on the vessel(s), this is then sent to the external software using the TCP interface. SAMS then calls the *receive* function. This function listens for data from the external software for the specified vessel (port) and receives and stores the data. To get the thruster forces for each thruster on the vessel, the *recieveForce* function is called from SAMS which returns the values received from the external software. These thruster forces are then applied on the vessel and stored in the output file before the loop starts again.

³[https://msdn.microsoft.com/en-us/library/windows/desktop/ms740673\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms740673(v=vs.85).aspx)

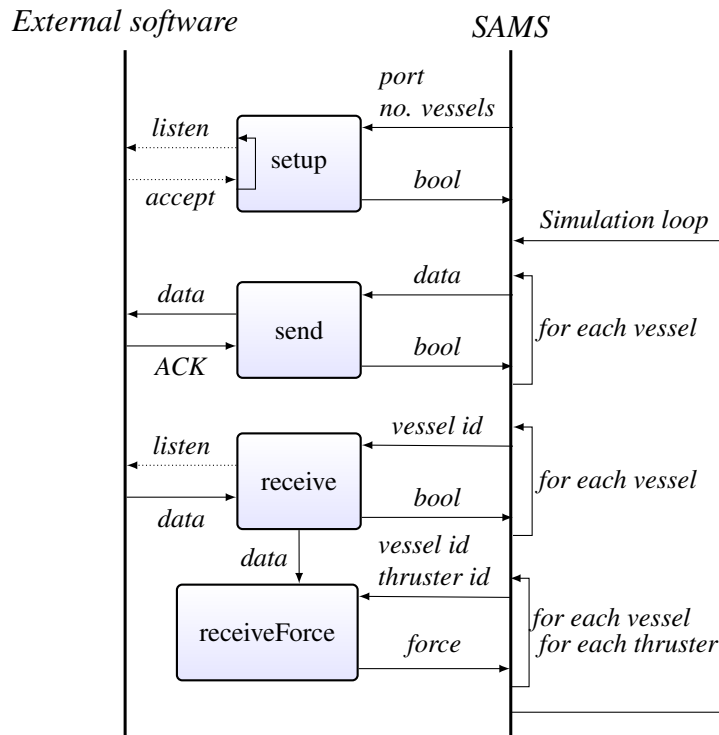


Figure 6.2: Detailed workflow of TCP interface from Figure 5.1.

6.2.2 Multiple Vessels

When using multiple vessels, each vessel gets its own port, and for simplicity the first vessel is initialized with the port specified in the *RHAS.itconfig* file. The subsequent vessels get the next ports, e.g. for three vessels where the specified port is 100 the port numbers will be 100, 101 and 102 respectively. Error handling to the cases when a port is occupied or out of the valid range has been added such that no illegal operations can occur. The connections and handling of multiple vessels have been handled sequentially, that means that if one vessel should disconnect or stop to send, the program will come to a stop. This solution is chosen both for simplicity and for data validity, running in parallel could have increased the performance a bit, but at a cost of additional failure handling.

6.2.3 Serialization Format

In the work of making the TCP library Simulink has been the main client and the server is as of this configured to suit Simulink. This means that the data is sent as 8-byte double precision values, without the use of serialization formats to handle the data. This means that the client and server need to have knowledge of the type and size of data that

is sent before the simulation starts. This way has been chosen due to its simplicity and limitations in Simulink that makes it hard and inconvenient to develop code with external C++ libraries and/or external TCP serialization formats.

One advantage to not using serialization formats is the reduction in size of the data that is sent. Depending on the format and information sent together with the data this could make up a huge amount of data over a long simulation and thus slowing down the simulation. Another consideration that needs to be taken is how far apart the simulator and the external software communicating through TCP/IP is on the network, as well as the speed of the network communication. If the simulator and external software is placed far away from each other on the network or the network speed between them is low⁴, more considerations should be put into the size of the data. Another advantage of sending raw bytes is the reduced encoding and decoding time as only the raw bytes and values needs to be encoded/decoded directly to values and raw bytes respectively, instead of also having to encode through a serialization format.

6.2.3.1 RapidJSON

Despite the advantages mentioned above with sending raw bytes with the main advantage being faster sending and interpreting of the data, it needs as mentioned above the knowledge of the type and size of the data. The more complex the data becomes the harder it is to keep track of, and backward compatibility might also become a problem when sending raw bytes. By using a serialization format the knowledge of the data is not needed, in addition the serialization format packages the data into a format that is easily readable for a human and a format that is well suited for storing.

The problems with sending raw bytes was solved by implementing a second method for sending and receiving data to the simulator over TCP. This method supports sending and receiving JavaScript Object Notation (JSON)⁵ objects using the rapidJSON⁶ coder. This method can easily be activated from the *RHAS.itconfig* file. As previously mentioned, using a serialization format increases the size and complexity of the data sent over TCP. Despite this a second method in the TCP interface was created utilizing the rapidJSON coder, this method however only supports one vessel, and for simulation purposes it is strongly advised to rather use the standard byte stream TCP implementation mentioned above. This method does not support the use of custom hydrodynamics as mentioned in Section 5.2.2, though this could easily be changed if needed. However this method pro-

⁴The same problem occur if the network utilization is high, even though the network speed (*bandwidth*) might be high and the network distance between them might be small.

⁵<https://www.json.org/>

⁶<http://rapidjson.org/>

vides an easy to use interface that have higher backward compatibility for the receiving and sending of data.

6.2.4 Failure Handling

As SAMS runs the main simulation loop, it is important that the TCP library does not shut down or stop to work even if a failure occurs. As mentioned earlier, TCP makes sure the information is sent and received as well as check the data for corruption, and resends the data if faults are found. However other faults might occur like failure to establish a connection or loss of connection. In those cases the TCP interface outputs an error message and send *false* back to SAMS indicating that the operation has failed in some way. It is then up to the SAMS code to take appropriate action and close down all connections. The ports used by SAMS are made available once SAMS closes down the library, either by failure handling or at the end of a simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

Implementation and Verification of Motion Control Systems

This chapter will introduce the motion control system implemented to work together with the simulation environment described in Chapter 5. This includes a motion control system implemented in Simulink described in Sections 7.1 through 7.6 and a Graphical User Interface (GUI) used for manual control described in Section 7.7.

7.1 Simulink Model

The Simulink model have been the main motion control system in this thesis and all simulations presented in Chapters 9 to 12 are carried out with the implemented Simulink model. The different parts of the model will be presented in detail in Sections 7.2 through 7.6, where verification and performance tests for some selected systems will be presented in the corresponding sections.

The Simulink model is connected to the simulation environment through a TCP send and receive block from the Instrument Control Toolbox¹ in Simulink. The TCP send and receive blocks connects to a specific port and IP-address specified by the user. Since no serialization formats are used, it is important to make sure the right amount of data is sent and received. More information about the setup of the connection and about the data format can be found in Appendix E.2.

The Simulink model itself is made to be as readable and easy to understand with a workflow from left to right and top to bottom, and color coding according to the block

¹<https://mathworks.com/products/instrument/supported/tcp-ip.html>

	Maximum
Velocity (surge)	8.75 m/s
Acceleration (surge)	0.09 m/s ²
Acceleration rate (surge)	0.003 m/s ³

Table 7.1: Saturation values for Oden in SAMS.

functionality. The color scheme used can be seen in Table F.1 and the main model can be seen in Figure F.1.

7.2 Reference Model

The reference model used in this thesis is the linear third-order reference model described in Fossen (2011), this model consists of a LP filter cascaded with a mass-damper-spring system, and the model can be seen in Figure F.2. As the reference model is of third-order the output is not only a position trajectory, but also a velocity and acceleration signal. The velocity and acceleration is converted to a BODY-fixed coordinate system before they are outputted. As described in Section 4.5, a drawback to the linear reference models is the change in response from different input amplitudes, one way to solve this as opposed to amplitude gain scheduling, mentioned in Section 4.5, is to saturate the velocities and accelerations.

In order to find the saturating limits an open water simulation was conducted in SAMS with a modified version of Oden presented in Section 8.1.1. In this simulation maximum thrust was requested from the thruster (dynamics) in surge, and the acceleration rate, acceleration and velocity was recorded, and can be seen in Table 7.1. The simulation is conducted only in surge under the assumption that any vessel moving in ice infested waters will try to move as much as possible in surge direction since sway and yaw will increase the resistance substantially, another assumption is that these values are equal in reverse as well as linear and valid for the full velocity range. Since the vessel will for the most part be in IM and DP where the velocity is low and reversing will not occur, this assumption should be valid.

During implementation it was noted that in order to obtain a satisfactory position and velocity trajectory, a rate limiter had to be added to the acceleration in addition to the velocity and acceleration limits. This ensures that the rate of acceleration is within obtainable rates for Oden in SAMS. The rate limiter prompted a slower LP filter so the input to the mass-damper-spring system was more in line with the saturation values.

	Surge	Sway	Yaw
Natural frequency ω_n	0.0571 rad/s	0.0571 rad/s	0.0571 rad/s
Relative damping ζ_i	1	1	1

Table 7.2: Natural frequency and relative damping for the reference model.

7.2.1 Setpoint Reference Model

The reference model from Fossen (2011) is used as a setpoint reference model where the input is a list of setpoints. These setpoints have been manually created, however they can also be coming from a path generation algorithm. The output is a position, velocity and acceleration signal that have been saturated according to Table 7.1, so any setpoint input should give obtainable output signals for Oden. The only parameters that require tuning of the model are the natural frequencies and relative dampings of the mass-damper-spring system (the LP filter gain is the same as the natural frequency), these values have been tuned to be perfectly damped with a period of 110 s and can be seen in Table 7.2.

Note that this reference model is only used in the development and verification of the controller and observer, where a constant list of setpoints have been given as input so the tuning could be performed in a consistent manner. When doing DP simulations the main motivation behind dropping the reference model is to speed up the simulation time as only a constant setpoint would be needed. As for the other simulations performed in this thesis, these prompted other modifications and other reference models and will be introduced in Sections 7.2.2 and 7.2.3.

7.2.2 Velocity Reference Model

The simulation presented in Chapter 10 consist of a transit through an ice field. As of such a simple reference model that outputs a constant velocity has been created, as can be seen in Figure F.3. This reference model was created with the intention that it should work together with the existing controller i.e. output a position, velocity and acceleration signal, but in turn makes the controller a virtual speed controller. The input is the desired velocity as well as the initial position of the vessel. The reference model assumes the vessel is at rest at initialization and as of this a second-order LP filter as well as a saturation on the acceleration is implemented to satisfy the dynamics of Oden.

The acceleration in the reference model was limited at 0.05 m/s, this is lower than the maximum acceleration of Oden, but is a good representation of the mean acceleration. It is also desirable to have the reference model slower than the vessel, guaranteeing good

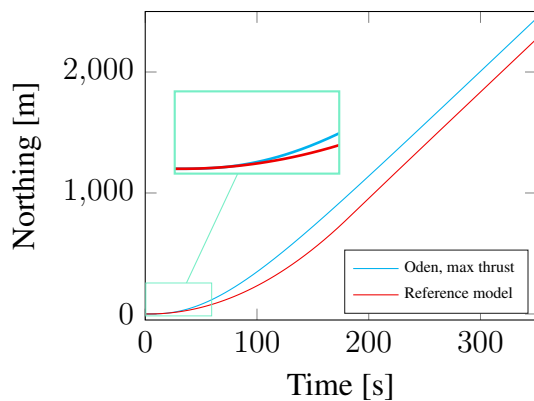


Figure 7.1: Position output from velocity reference model and actual position of Oden.

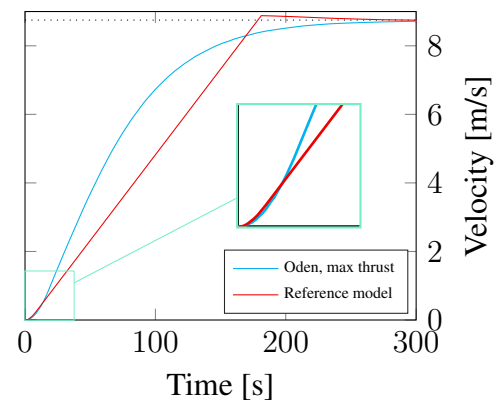


Figure 7.2: Velocity output from velocity reference model and actual velocity of Oden.

tracking performance over fast response. The second-order filter is chosen because it will have a slower incline at the start and end, but at the same time have a steeper incline in the middle, making a fast yet manageable move toward the desired velocity.

7.2.2.1 Verification of the Velocity Reference Model

As a verification of the velocity reference model, a simulation was performed where the input to the velocity reference model was the top speed of Oden at 8.75 m/s, and the results can be seen in Figures 7.1 and 7.2. Here it can be seen that the position is the integrated velocity and that the velocity has a constant slope as a result of the saturation. In the figures a simulation of Oden where maximum thrust in surge is requested from the thruster dynamics, bypassing the controller, from a complete standstill is also shown. It can clearly be seen that Oden have faster dynamics than the reference model in the lower velocity range, but towards the higher velocities have problems keeping up with the reference model due to the high hydrodynamics forces. This is as designed and expected since simulations rarely would be carried out at top speed, and therefore the reference model is considered to be fully functioning.

7.2.3 Ice Management Reference Model

The major change to the reference model used for IM is that the path is coming from a script written to mimic an arched racetrack based on the results from Holub et al. (2018), see Figure 7.3. The path is generated using MATLAB, and is giving a high density of waypoints as output in the arched racetrack pattern. As the ice condition makes it hard to perfectly track the pattern and the need for a perfect tracking is not present, the error distance for when to move to the next point was put to 20 m. This

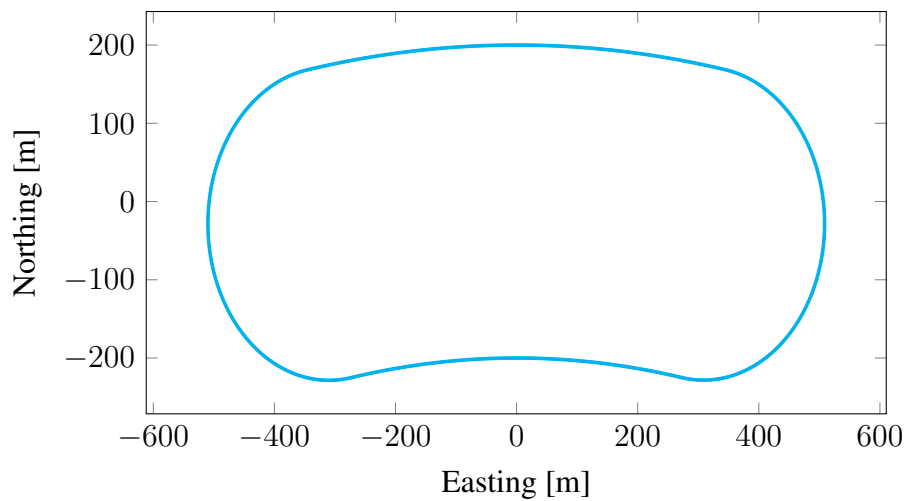


Figure 7.3: Arched racetrack path used in the IM reference model.

speeds up the controller and gives a smoother path, as the difference in the current and the next setpoint will be small, independent of where the vessel is positioned. The Simulink model for the IM reference model can be seen in Figure F.4.

In addition to the modification above, the reference model also check if the next setpoint in the list is closer than the current setpoint, if this is the case it means that the vessel have gone past the setpoint, but not been within the error distance. In these cases the reference model will set the next setpoint as the current setpoint to avoid the need for turning back.

A big error in sway position might give unwanted behavior and this is solved by calculating the desired heading as the angle between two points approximately 100 and 200 m ahead in the path. This results in a modified Line of Sight (LOS) steering law, the big lookahead distance is chosen because it results in good performance based on the ice-conditions and desired velocity of the IM vessel. It is thus up to the sway compensator, which will be introduced in Section 7.4.3, to reduce the sway error.

The other big modification on the IM reference model is that the velocity calculated by the reference model is not used, instead a constant speed is outputted in surge, as this is the speed the IM vessel should hold through the ice field. The desired velocity is also reduced when going with the current making the downstream turn on the West side of the path easier as well as eliminating unnecessary ice-breaking in this part of the path.

7.3 Observer

The nonlinear passive observer described in Section 4.6.1 is used as an observer in this thesis, and the Simulink model can be found in Figure F.5. The model is moderately modified, the biggest modification is that the first-order wave estimator is excluded from the observer, as there is no waves in SAMS. Since the vessel is operating in ice, the ice forces have been included in the observer as they can be assumed known based on the theory in Section 3.2. Another big modification is the inclusion of the hydrodynamic model described in Chapter 3.6 for the dynamics of the vessel. This can be seen as unrealistic since it is not always possible to have such a good estimate of the dynamics of a vessel, but as it is available and should be possible to do for any vessel² it is utilized. The observer outputs an estimate of the position, velocity and acceleration, where the acceleration signal is estimated by dividing the sum of all forces and observer gains with the vessel mass.

To be more realistic, white noise have been added to the position coming into the observer. The noise is to mimic noise and uncertainties from any GNSS-signals, and the amplitudes have been set to follow the accuracy of a Differential Global Positioning Systems (DGPS) signal which is approximately 0.1 m. Since the update rate on GNSS-signals is not equal to the update rate on most controllers, a zero-order hold with a time constant of 1 s is also added so the signal to the observer is as realistic as possible. The noise is filtered with the same LP filter used in the reference model and the gains for each filter can be found in Table 7.3a. Noise is also added to the ice forces since the simulator returns the actual forces and in any realistic situation these can only be estimated according to Section 3.2.

The tuning of the observer is performed as mentioned with the setpoint reference model and a constant setpoint list. The observer gains have been tuned quite aggressively to get a good estimation of the signals, and the gains can be seen in Table 7.3b. One concern with the aggressive tuning is that some noise might be carried forward to the controller. This has been partly solved by the introduction of the LP filter to try and remove as much noise as possible before the observer. The LP filter has been tuned to minimize the signal delay while trying to suppress the signal noise as much as possible.

7.3.1 Verification of Observer

The estimated position and velocity for a simulation with the setpoint reference model and the standard controller can be seen in Figures 7.4 and 7.5. Notice here that as a

²System matrices has been generated using SINTEF'S ShipX

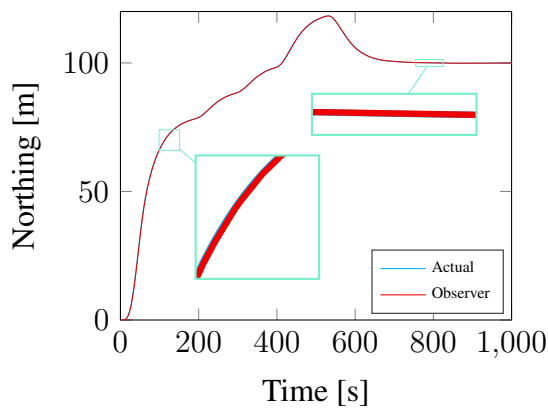


Figure 7.4: Actual and estimated northing positions from the observer with white noise.

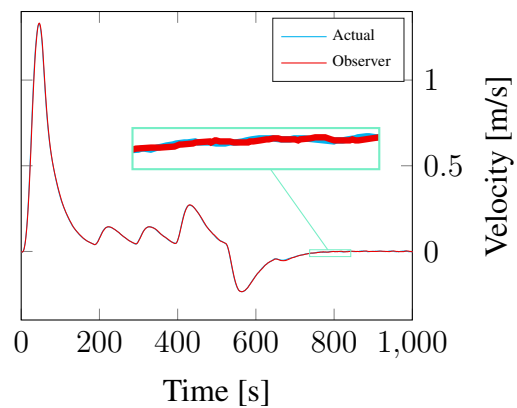


Figure 7.5: Actual and estimated surge velocities from the observer with white noise.

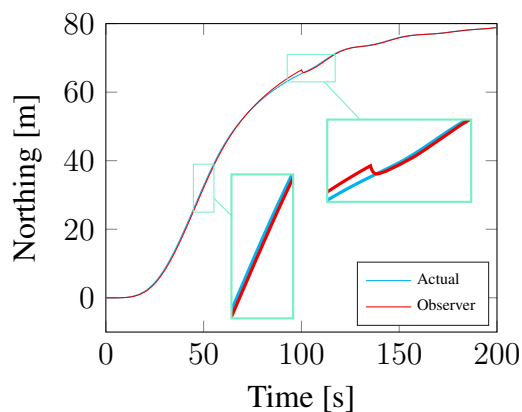


Figure 7.6: Actual and estimated northing positions from the observer during loss of signal simulation.

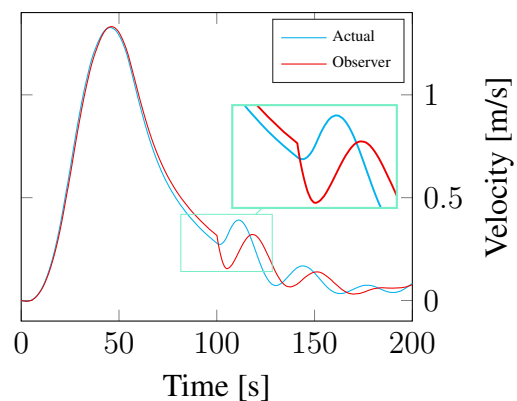


Figure 7.7: Actual and estimated surge velocities from the observer during loss of signal simulation.

result of the LP filter on the noisy signal a delay is introduced into the observer, this is especially noticeable on the velocity signal in Figures 7.5. Note that Figures D.1 and D.2 presents the same signal with the setpoint reference model, but only in the time period between 770 s to 830 s.

As a more extreme test of the robustness and stability of the observer a second test was simulated where the added noise was 20 times bigger, this is presented in Figures D.3 and D.4. This increase in noise amplitude is done to check if the observer and LP filter can suppress it, this big increase is somewhat unrealistic as it can be seen as twice the inaccuracy of a normal Global Positioning System (GPS) signal. However it is a good test of the observer and it shows that the filter and observer cannot suppress the noise and the noise is passed through to the controller resulting in a velocity with lots of noise as can be seen in Figure D.4.

	Filter gain
Surge	2.51
Sway	2.51
Yaw	0.52

(a) Low Pass filter gains.

Gain	Value
T	$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
K_2	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$
K_3	$\begin{bmatrix} 9 \times 10^5 & 0 & 0 \\ 0 & 1 \times 10^6 & 0 \\ 0 & 0 & 1.2 \times 10^8 \end{bmatrix}$
K_4	$\begin{bmatrix} 6 \times 10^5 & 0 & 0 \\ 0 & 1 \times 10^6 & 0 \\ 0 & 0 & 6.5 \times 10^8 \end{bmatrix}$

(b) Values for observer gains.

Table 7.3: Gains and constants for the observer.

A third test was also simulated to test the observer, namely a simulation of a loss of input signal. The simulation was run for 50 s when the northing position signal was removed, then the simulation was run for another 50 s before the signal was restored. The position and velocity from the observer compared to the actual position/velocity is shown in Figure 7.6 and 7.7. Here it can be seen that the position is deviating with about 2 m after 50 s of loss of signal and the velocity is deviating with 0.04 m/s. Note here that no filtering of the input signal is done so when the signal is turned back on the observer sees a big error resulting in large error signals from the observer gains. This is what can be seen in Figure 7.7 right after the signal is restored, and an instability is then introduced. This is as expected and it can be seen that the observer after some time manages to remove the instability. In reality this sudden amplitude increase of the input signal should not occur as some type of signal merging and signal filtering should be implemented.

7.4 Controller

The controller chosen for this thesis is a PID controller, this is mainly due to its simplicity in implementation and tuning, but also because it is used in many variations in the industry. The Simulink model can be found in Figure F.6. The PID controller itself is a decoupled PID controller consisting of three independent controllers each controlling

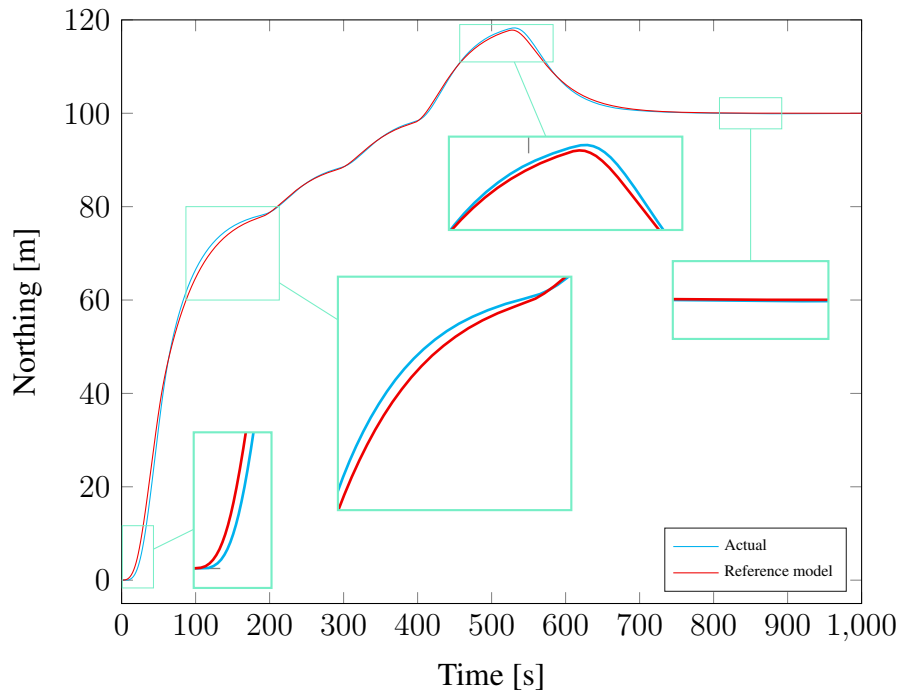


Figure 7.8: Tracking performance of the controller compared to the setpoint reference model.

the surge, sway and yaw motion of the vessel, where the input to the controller is the error in position and velocity. This error is calculated based on the outputs from the reference model described in Section 7.2 and observer from Section 7.3, where the error is defined as the reference model signals subtracted by the observer signals. The velocity error is used in the derivative term of the controller, making the controller more stable.

The tuning of the controller is performed with the same setpoint reference model and ice field as used to tune the observer. The tuning is based on Algorithm 1, however the ice field prompted a more aggressive tuning of the controller in order to achieve a satisfactory tracking. The final values of the controller gains can be found in Table 7.4.

7.4.1 Verification of the Controller

A list of setpoints was given to the reference model and the actual signal of the vessel compared to the reference signal was plotted, this can be seen in Figure 7.8. Here the fact that the vessel is in an ice field can be noted by the slower response in the beginning, it can also be noted that the more aggressive tuning makes the controller overshoot slightly but reaches steady state with zero error.

Gain	P	I	D
Surge	$1.9K_p$	$3.0K_i$	$2.0K_d$
Sway	$1.2K_p$	$2.0K_i$	$1.2K_d$
Yaw	$1.2K_p$	$2.0K_i$	$1.9K_d$

Table 7.4: PID tuning values.

7.4.2 Acceleration Feedback Controller

An acceleration FB term is also added as an option to make the controller more responsive to external disturbances such as the ice forces, this was mainly done so a comparison could be made between a controller without acceleration FB and one controller with acceleration FB. The acceleration FB controller can be seen in Figure F.7. The acceleration signal is estimated by the observer and summed with the acceleration signal from the reference model to get the error, then it is passed to the controller and is reduced to 0.6 to 0.8 of its original value before it gets added. The signal is then multiplied with the mass of the vessel to get a force and is then summed with the other force demands coming from the controller.

7.4.3 Sway Compensation

A *compensate sway* block is added to the yaw controller to mimic a basic LOS controller. The main motivation behind this is that when navigating in ice any sway error should be compensated with a change in heading as the vessel moves forward, i.e. minimizing the movement in sway direction. The compensate sway block is activated when the speed of the vessel is above 1.5 m/s, i.e. not in DP mode, making this only activate for the transit and IM simulations.

The compensate sway block works by using the error in sway position, \tilde{y} , and adding a calculated yaw angle to correct for this sway error over a certain distance along the path, d_x , according to (7.1). E.g. for the DP compensate block the path distance is 15 m, meaning that the sway error is zero after 15 m along the path.

$$\psi_{comp\ sway} = \tan\left(\frac{\tilde{y}}{d_x}\right). \quad (7.1)$$

7.4.4 Ice Management Controller

As the vessel doing the racetrack path in the IM maneuver needs to have a certain speed relative to the ice drift velocity in order to maximize its efficiency, the controller for this

vessel was modified to control the surge velocity. The surge controller is changed to take the error in velocity instead of position as input making it a virtual speed controller. This solution is not ideal since it is quite aggressive, but it is chosen since the most important aspect with the IM vessel was that it could follow the path with a certain speed. In reality the IM vessel is manually controlled allowing it to do more fine maneuvering and also reverse if it get/are about to get stuck. In addition other minor changes have been implemented to the IM controller, and the controller can be seen in Figure F.8.

7.5 Thruster Allocation

The thruster allocation was made using the pseudo inverse strategy from Section 4.3, where the the force coefficient matrix \mathbf{K} is multiplied with the thrust configuration matrix. The model can be seen in Figure F.9. As this method is not a smart algorithm, a weighting matrix \mathbf{W} was added to make the behaviour of the vessel's thrusters more realistic. The weighting matrix tries to minimize the use of the bow thrusters, especially at high speeds. One assumption is made on the azimuth thrusters and it is that they can be split into two different thrusters giving a force in only surge- and sway-direction respectively.

In the thrust allocation three different scenarios was created, where both the weighting matrix \mathbf{W} and the thrust configuration matrix \mathbf{T} was changed in order to achieve the correct behaviour. The three different scenarios are:

- *DP*: Speed of the vessel is below 1.5 m/s. Here no limitations are put on the thrusters.
- *Low-speed maneuvering*: Speed is below 3 m/s. Here the weight matrix is set to penalize the use of the bow thrusters in order to minimize its use, and also the thrust configuration matrix is tuned to only allow symmetric thrust in the surge direction of the azimuth thrusters to generate zero yaw-moment.
- *High-speed maneuvering*: Speed exceeds 3 m/s. Here even more restrictions are put on the bow thrusters making them virtually impossible to use, also the weight on the sway direction of the azimuth thruster are penalized a bit more than the surge direction.

7.5.1 Thruster Dynamics

When modelling the thrusters their dynamics have also been considered. This means that a delay has been put on each of the thrusters. For simplicity the azimuth thrusters

are assumed to be limited by its turning rate of $5.0^\circ/\text{s}$ which when turned into force equal a rate change of $33\,333.3\text{ N/s}$ in the thrusters surge and sway directions. Note that the rate change of the motor is not taken into account as this is considered to be faster than the turning dynamics. This though lead to a somewhat slower response in change of thrust than what is realistic. One other assumption is that the propeller and thruster efficiency is the same for all water and propeller speeds, thus making the power to thrust ratio constant at 0.1 N/W .

For the bow thrusters they have been modelled to have a dynamic satisfying zero to full power in 10 s. As well as with the azimuth thrusters the bow thrusters are assumed to have a linear response and a power to thrust ratio of 0.1 N/W .

7.6 Hydrodynamics

As hydrodynamic forces are small compared to ice forces, the modeling of hydrodynamic forces have not been put much attention to in SAMS. As a result of this an extended hydrodynamic model have been added to Simulink as part of this thesis. The hydrodynamic model is based on Perez and Fossen (2007) and includes velocity dependent fluid memory effects, cross-flow drag and surge resistance (quadratic drag) and linear drag as presented in Section 3.6. The Simulink model used for the hydrodynamics is modified from the MSS toolbox, Fossen and Perez (2004), and can be seen in Figure F.10. This model requires system information for Oden, as described in Sections 3.6.2 and 3.6.3, and these have been calculated by Jon Bjørnø using ShipX.

Some modifications was implemented in the model for use in this thesis, the hydrostatics was removed as this is calculated in SAMS as well as removing the fluid memory effects in pitch as this was causing instability. The reason for the instability was never discovered, but was most likely an effect caused by the fact that the hydrostatics are calculated in SAMS. For more realistic results an exponential term is added to eliminate the linear drag at high velocities. This can be added by assuming that the quadratic term is much bigger than the linear term at high velocities.

7.6.1 Tuning of the Model

The Simulink model from MSS needs tuning since the original values for linear and quadratic drag results in a top speed of Oden much lower than the assumed top speed of 17 kn. By assuming that the linear drag is zero at high velocities and assuming a maximum thrust in surge of 1.2 MN , the quadratic drag coefficient C_x can easily be calculated from the quadratic drag equation in surge (3.56). The linear damping B_v in

Model	Parameter	Value
Simulink	C_x	0.1474
	$B_{v,surge}$	0.3242
SAMS	<i>waterFormDragCoefficient</i>	0.0167
	<i>waterSkinFrictionCoefficient</i>	0.0005

Table 7.5: Parameters used in hydrodynamic models.

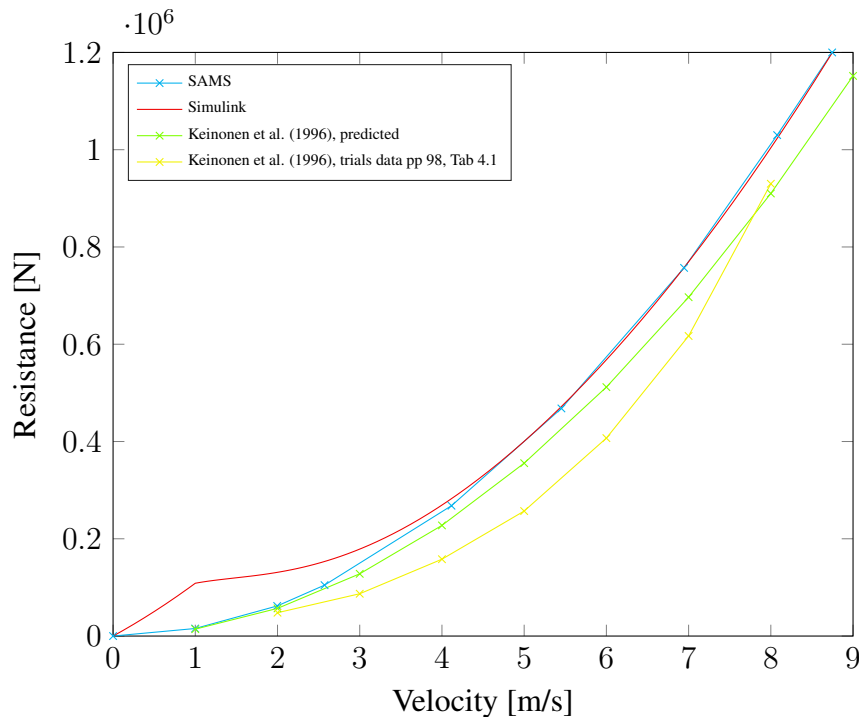


Figure 7.9: Comparison of hydrodynamic resistance for Oden.

surge is now tuned by assuming C_x is one at 1 m/s thus making the linear damping at 1 m/s the difference in quadratic drag with the different values for C_x , resulting in the values found in Table 7.5. The resulting hydrodynamic force can be seen in Figure 7.9. In the figure a tuned version of the hydrodynamics in SAMS can also be seen, this is tuned to a top speed of 17 kn with coefficients as seen in Table 7.5.

7.6.2 Verification of Hydrodynamics

The SAMS hydrodynamic forces in Figure 7.9 are tuned to a top speed equal to the Simulink model and therefore it is as expected that these two coincide for large velocities where the linear drag term is zero. Keinonen et al. (1996) gives a formula for predicting the open water resistance of vessels and this can also be seen in Figure 7.9. The predicted values by Keinonen et al. (1996) are a bit lower than the SAMS and

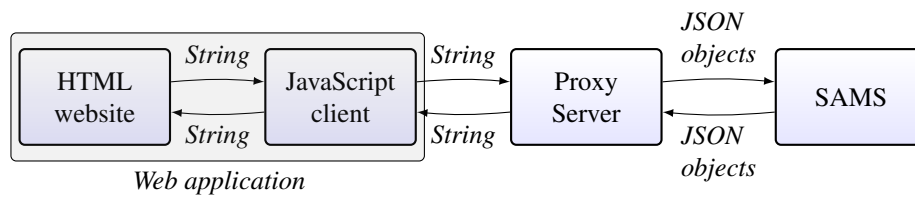


Figure 7.10: Web application signal flow.

Simulink models for higher velocity, but much lower for lower velocities compared to Simulink. This may imply that the linear damping of the Simulink model is too high. Note that the prediction formula is a best fit curve of the resistance from several sharp chinned icebreakers. Keinonen et al. (1996) also gives data from sea trials with Oden and the results on the open water resistance is seen in Figure 7.9, this clearly shows a lower resistance than the other models. This can be from how the resistance is calculated, any wind, current or other factors not accounted for or ballast condition for the vessel during the test. The fact that the trials are considerably lower than the theory is something worth investigating closer, however this is out of scope for this thesis.

7.7 Manual Control

In addition to the Simulink model introduced in the sections above, an application to manually control a vessel in SAMS was created. The application was made to work with the TCP interface described in Section 6.2 and specifically to work with the rapid-JSON serialization format version of the interface. Doing it this way opened up the possibility to create a HTML website with controllers such as joysticks and slider bars each controlling a thruster. Note that the current setup of this web application is only for controlling a single vessel and the vessel is the modified version of Oden, details on the modified vessel can be found in Section 8.1.1. The web application also contains a JavaScript client which handles the communication with a proxy server, and the setup with a high level signal flow is seen in Figure 7.10.

7.7.1 Website

The website was created using HTML and works as the GUI, the joysticks and slider bar was created using JavaScript and linked with HTML to the JavaScript client. As this is the GUI it was designed to be simple and easy to use, while also providing its functionality. The position and velocity values are displayed at the bottom of the page and is updated every time a new data-package is received from SAMS. A screenshot of the website is given in Figure 7.11.

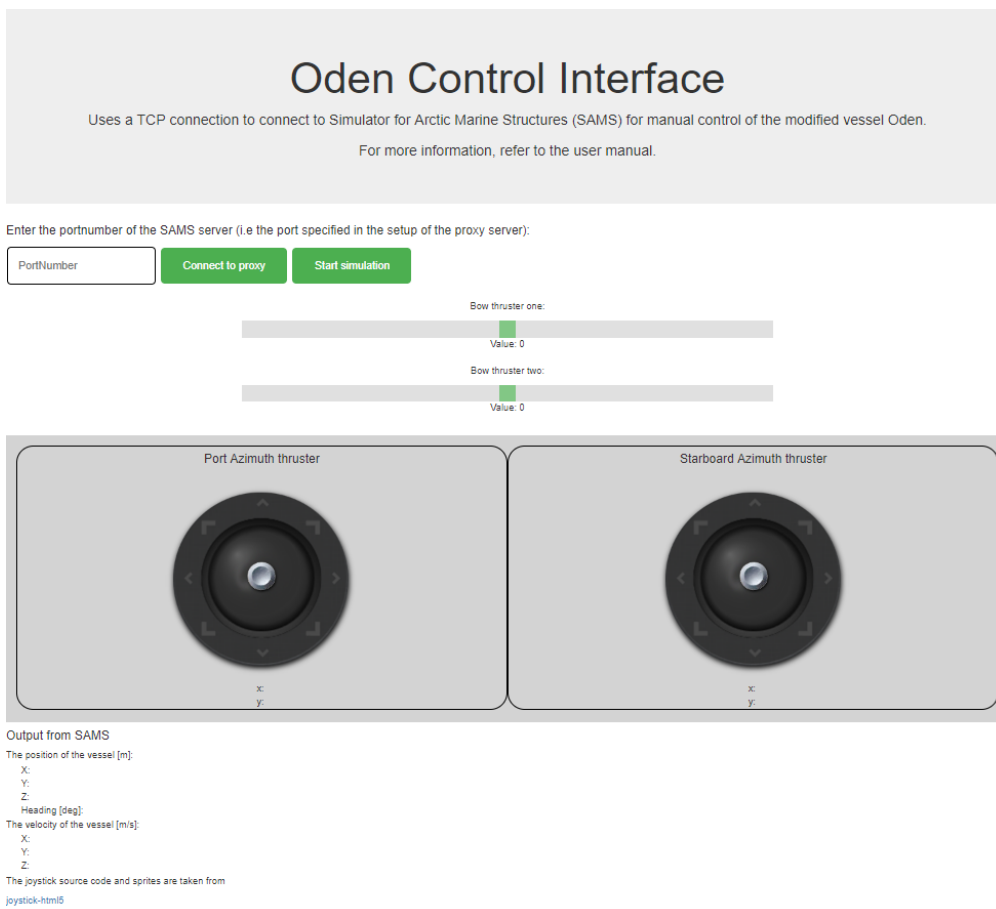


Figure 7.11: Website interface for manual control of Oden.

	Time constant
Bow thruster	10 s
Azimuth thruster	30 s

Table 7.6: Time constants for LP filter in proxy server.

7.7.2 JavaScript Client

The JavaScript client is made in the web application to enable communication of the position of the controllers. These positions are read by the website and transformed to strings before the client sends them to the proxy server for interpretation and encoding. The JavaScript client also receives the position and velocity of the vessel from the proxy server as a string and then splits it into the specific values so it can correctly be visualized on the website.

7.7.3 Proxy Server

JavaScript does not in its pure form support rapidJSON, therefore a *backend* proxy server was made using Node.js³. The proxy server is responsible for connecting the JavaScript client of the web application to the TCP interface. The proxy server is receiving JSON objects containing the position and velocity of the vessel from SAMS, and is responsible for decoding the data and restructure it in a string so it can be sent to the JavaScript client. The proxy server also receives a string which contains the position of each controller on the website and maps these positions into thruster forces before encoding it in a JSON object and sends it to SAMS.

Since the JSON objects are sent with TCP between SAMS and the proxy server these do not require to be started on the same computer, however the proxy server and the web application needs to be run on the same machine as the proxy server are set to listen on the local network for the JavaScript client.

In order to model thruster dynamics a LP filter was added on each thruster to simulate some delay in the forces, the LP filter was tuned so the delay on the azimuth and bow thrusters are according to Table 7.6. This was done to make the application more true to reality and also adding some features showcasing the possibilities of such an application. Notice with this configuration of the LP filter the amplitude gain scheduling problem occurs, and for realistic results this should be accounted for.

³<https://nodejs.org/en/>

Simulation Setup

This chapter will first introduce the general setup for the simulations and introduce the vessel model of Oden used for the simulations. Secondly a discussion on the improvements and assumptions performed on the vessel model will be presented. The specific setup, results and discussion for each simulation performed are presented in Chapters 9 to 12.

8.1 General Setup

All simulations was run with a version of the SAMS code with the TCP library implemented. The SAMS code used was the latest developer edition as of April 2018 with addition of a beta version of an improved fracture handler. The simulator configuration file, *SAMS.itcofing*, can be found in Appendix G.1. The file was constant during all simulations for this thesis with the exception of placement of the vessel, towing carriage properties, minimum floe mass and the current velocity vector. The *waterSkinFrictionCoefficient* and *waterFormDragCoefficient* was ten times higher than that in the hydrodynamic simulation in Section 5.2.2, as this has proven to give a better and more realistic result for the ice floes. Naturally this gives wrong hydrodynamic drag for the vessels, as of such the hydrodynamics was in all cases calculated by Simulink.

In the Simulink model all parameters were as described in Chapter 7. The reference model and controller were changed for the different simulations, information on this can be found under the setup for each simulation case.

All ice forces for SAMS have been filtered before the results were presented. The filtering technique used is a mean filtering technique, and the filtering have been done independently on the rigid and crushing ice forces. Based on recommendations from

ArcI So the results have been filtered over 20 s for the rigid ice forces and 1 s for the crushing ice forces. The main reason for the longer filtering of the rigid ice forces were that they are not physically correct compared to reality in the way that their forces are interpreted in the simulator. In reality the rigid forces compromises bending, buckling, sliding, submersion, shearing and splitting and can occur at the same time, while the simulator with its discrete state needs to apply the forces in a specific way and order. However the mean rigid force can be interpreted as a correct force. One other reason for filtering is that the vessels are modelled as rigid bodies in SAMS, meanwhile a flexible vessel will distribute the forces more evenly.

8.1.1 Vessel

The vessel Oden is in reality equipped with two Controllable Pitch Propellers (CPPs), and a vessel with this configuration is not intended as a DP vessel. Since this thesis simulates a DP operation in Arctic waters, modifications of Oden was made for use in SAMS. The modifications involve substituting the two CPPs with two 6 MW azimuth thrusters¹ providing up to 0.6 MN of force each. In addition two 1 MW tunnel thrusters were added in the bow of the vessel, this was done to make sure enough power is available to maintain DP capability during harsh ice conditions.

All the thrusters have been carefully picked with regards to both size and reasonable power yields. The total power for the main engines is substantially lower (12 MW) than the original configuration with four engines supplying up to 18 MW of power to the two CPPs. However the azimuth thrusters gets their power directly from an electric motor while the CPPs will experience substantial losses on the way. Furthermore a propulsion efficiency of 0.1 N/W is assumed for all thrusters.

Vessel data was collected from the vessel model created by Jon Bjørnø, vessel drawings and Keinonen et al. (1996). The relevant data can be found in Table C.1 and drawing of Oden with the modified thruster placements can be found in Figure C.1.

8.2 Discussion of the Modifications and Assumptions of Oden

The modifications to the Oden simulation model have been carried out in a careful manner by choosing thrusters that are reasonable for use in an Arctic environment, the

¹Information on the azimuth thrusters are taken from ABB: <http://search.abb.com/library/Download.aspx?DocumentID=9AKK105152A2911&LanguageCode=en&DocumentPartId=&Action=Launch>

azimuth thrusters themselves are selected from ABB among thrusters commonly found on icebreakers. This resulted in two azimuth propellers capable of giving an output of 1.2 MN of thrust, this seems to be correct at a top speed of 17 kn according to Keinonen et al. (1996). However it should be noted that it is probably far from correct for lower velocities. The specified bollard pull of the real Oden is 250 t^2 which is slightly more than double that of what the modified simulation model of Oden can provide of thrust. This can be seen to be caused by the constant propulsion performance of 0.1 N/W which is not physical correct. This should be dependent of the velocity of the vessel, however this is outside the scope of this thesis and since most of the simulations are within the maximum thruster output of 1.2 MN, this would not give any major change in the results.

The most unrealistic modification was the inclusion of bow thrusters, this was as a result of providing more sway force for when Oden would stay in DP. Tunnel thrusters are more scarcely found on icebreakers, but there exists ice going vessels with a similar thruster configuration as the modified Oden used in this thesis, e.g. the icebreaker Fennica³. There are two main reasons for modifying Oden rather than using a model of another vessel such as Fennica. The first being that Oden is the only icebreaker that has a high quality vessel model available for use in SAMS. The second reason being that Oden is a research vessel where a lot of full-scale data is available.

Oden has two systems of interest that have not been modeled in this thesis. One is the heeling pump which allows the vessel to roll from one side to another and thus help break up the ice in harsh conditions. The modeling of the heeling could easily be achieved by sending in additional forces with the hydrodynamic forces, however the use of the heeling system would require a smarter control system and the system would thus only be useful for manual interactions. The fact that all the simulations in this thesis is performed in uniform and fairly easy ice makes the use of the heeling ability unnecessary.

The other system that has not been modelled is the water lubrication system. This is a system that pumps water at high speeds out of nozzles at the bow of the vessel, this significantly reduce the friction between the ice and the vessel and can also to some degree be used as a bow thruster. However the modeling of the system would cause the need for additional computations and it is doubtful that the full effect of the system could be modelled in a good way. The system would also be most efficient at low velocity with snow covered ice as this increases the ice-vessel friction considerably, however snow is not modelled in the simulator and as of such the use of the water lubrication system is small.

²<http://sjofartsverket.se/pages/41381/Oden%20f%C3%B6r%20webben.pdf>

³<http://arctia.fi/en/ship/fennica/>

THIS PAGE INTENTIONALLY LEFT BLANK

Validation of SAMS in Terms of Level Ice Resistance

This chapter will present a simulation used to validate the level ice resistance of SAMS against empirical ice resistance models. First the setup of the simulation including the ice field used will be presented before the results from the simulation and the empirical ice resistance models will be presented. Lastly the results from this simulation will be analyzed and discussed.

9.1 Setup

As presented in Section 3.1, there exists multiple resistance models for level ice. For the validation of the simulator these models have been compared with a simulation inside a level ice field. The results were taken from a 2000 m simulation starting 100 m inside the level ice field, from this a mean value was calculated to get an estimation of the ice resistance.

For the level ice simulations a rigid towing carriage, i.e. no modifications from the original SAMS code, was used to get a constant velocity through the ice field. The simulations were completed with velocities varying in surge direction from 1 m/s to 5 m/s. Hydrodynamic forces were removed from the simulation results to be able to compare the results with the ice resistance formulas. The ice and water parameters used in the formulas can be found in Table 9.1, while the vessel parameters can be found in Table C.1, additionally it is assumed that Oden is a sharp chined vessel.

h	Ice thickness	1 m
σ_f	Flexural strength	5×10^5 Pa
E	Young's modulus	5×10^9 Pa
ν	Poisson's coefficient	0.3
ρ_{ice}	Density of ice	900 kg/m ³
ρ_{water}	Density of water	1005 kg/m ³
μ_{si}	Structure-ice friction coeff.	0.15
h_s	Snow thickness	0 m
t	Ice surface temperature	-2 °C
C_h	Hull condition factor	1.0
C_s	Salinity of water factor	1.0

Table 9.1: Ice parameters used in calculations by the level ice resistance models.

9.1.1 Ice Field

The level ice field is created with IceMaker. The level ice field is approximately 3000 m long and 2000 m wide and has a thickness of 1 m. Additionally the ice field was put between the tank walls such that no horizontal movement of the ice field was possible.

9.2 Results

Figure 9.1 shows the predicted ice resistance from each model together with the results obtained from the SAMS simulation. In addition to the SAMS version described in Chapter 8, one simulation was conducted in the same ice field with a version of SAMS without the beta fracture handler. Note that the SAMS version without the beta fracture handler are labeled as *SAMS master* in Figure 9.1 and that this is the only simulation in this thesis carried out with this version of SAMS. For the ice resistance models all values used in the calculations can be found in Tables 9.1 and C.1, and all models and results from SAMS are given without hydrodynamic forces.

9.3 Discussion

For the level ice simulation results presented in Section 9.2, it can be seen from Figure 9.1 that the values from SAMS gives slightly lower results than the ones found in SAMS master. It can also be seen that the results from SAMS gives slightly more linear results which are more in line with the resistance models. This suggests that the beta fracture handler implemented gives SAMS more accurate results.

The most advanced model, Keinonen, which among others used Oden as a basis for the

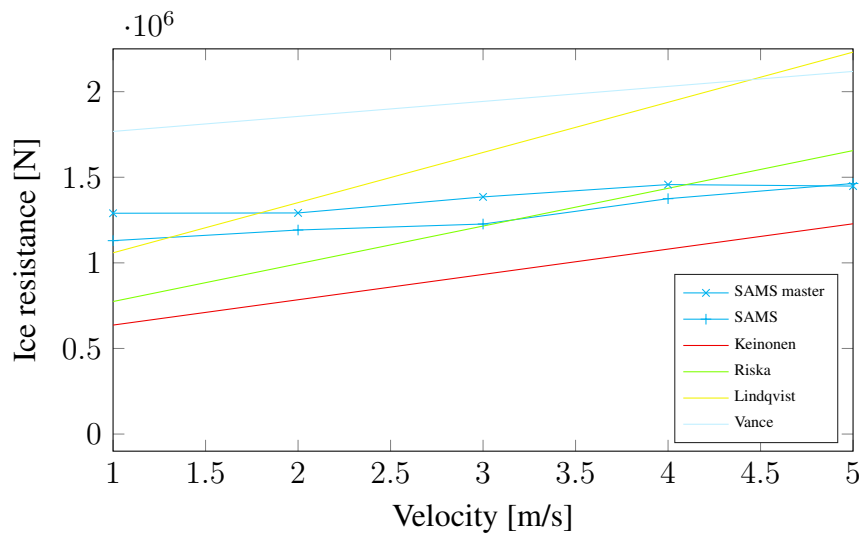


Figure 9.1: Predicted and simulated ice resistance in level ice.

prediction formula predicts the lowest resistance. It should be noted that it is stated that the *hull resistance increase coefficient* used in the calculation of the Keinonen model is only 0.009, and Keinonen et al. (1996) stated that this often needs to be increased, up to a factor of 3, for vessels with a deep wedge. Oden however only has a small wedge and let most of the ice floes move without much direction change, Keinonen et al. (1996), however this coefficient should likely have been put a bit higher for the calculation. The hull condition factor¹ is also assumed low with a value of 1, which indicates a completely new and freshly painted hull. For bare steel a factor of 1.33 is often recognized, implying that a higher coefficient will often be the case, especially for ice going vessels.

As the Riska model is partly based on the Lindqvist model and calibrated against several icebreakers, it is natural that the Riska model gives results more in line with SAMS compared to the Lindqvist model. However the Riska model is a simplified model and only the ice thickness is used as a parameter describing the ice, while the stem angle is the only parameter describing the bow shape. As a result of these simplifications the model is unreliable for different ice conditions and bow shapes. Note that values for the flexural strength and frictional coefficient are incorporated into the constants used in the model.

Erceg and Ehlers (2017) reports that the Lindqvist method does not produce reliable results, and it can be seen that this model gives the highest slope of all the models. The large slope is a result of an unreliable velocity dependent term together with an

¹Used to indirectly change the structure-ice friction coefficient.

unreliable crushing term in the model. The crushing term is highly reliable on the bow flare angle, as such this may imply that the model is unreliable for large bow flare angles like the one found on Oden.

Vance shows a substantially higher resistance than both the SAMS simulations and the other models, this can be due to the fact that the model is based on a single set of trial data. This makes the model unreliable for predictions where the shape of the vessel or ice conditions differ from the ones the model is based on. The model is based on data with a relatively high flexural ice strength and a fairly deep snow cover, this gives additional resistance making the total resistance higher than expected and could explain at least part of the high predicted results.

The Lindqvist and Vance models are seen to predict unreliable results for Oden, as such they can not be used to verify the correctness of the simulator. The Riska and Keinonen models give results in the same range as the simulator, but with considerably less ice resistance at low velocities. The uncertainties in the parameters used explains the difference between the two models and more correct values would likely put them closer together.

It should be noted that the simulation in SAMS was carried out with a rigid towing carriage. This can be seen as highly unrealistic as no vessel would have the power requirement to keep the velocity constant in a level ice field. As of this it is natural that the incline of forces in SAMS are lower than what is seen in the empirical models. Without a rigid towing carriage the vessel would have a bigger fluctuation in velocity and would likely get the possibility to slide on top of the ice and initialize bending failures in a bigger degree than what have been possible in the simulation. This could explain part of the low slope of the simulator results.

SAMS was not designed with level ice in mind and the results must therefore be treated accordingly. The large number of unknowns also makes it hard to determine which of the results are most correct. However, SAMS seems to give satisfactory results in line with both the Riska and Keinonen model, though more work on the simulator and tuning of simulation parameters seem necessary.

OATRC 2015 Transit Simulation

The OATRC 2015 transit simulation will be presented in this chapter. Firstly the setup of the simulation and the ice field used will be presented, before the results from the simulation with predicted resistance from the Croasdale model and a modified Croasdale model will be presented. Lastly the OATRC 2015 simulation will be compared, analyzed and discussed with regards to results from full-scale trial data as well as the Croasdale model and results from Kjerstad et al. (2018).

10.1 Setup

A simulation representing a transit during OATRC 2015, more specifically 30th September 2015 between 6.49AM and 7.35AM, was carried out to verify high velocity capabilities of the simulation environment and the Simulink motion control system. The simulation purpose was to mimic the data from OATRC 2015, hence the TCP interface was activated in SAMS and a free moving vessel controlled by Simulink was used.

For the OATRC transit real propulsion and position data have been available, as of such the simulation was compared to a portion of the full-scale data between 7.05AM and 7.09AM. The ice forces from the full-scale transit was calculated using the shaft power and velocity according to Juva and Riska (2002). During the full-scale transit a mean velocity of 5.69 m/s was recorded, with minimum and maximum velocity ranging from 2.5 to 6.9 m/s. The simulation was carried out with the velocity reference model described in Section 7.2.2 with an input velocity of 5.69 m/s. In addition the sway compensator was reduced to $\tilde{y}/1000$ to allow for more error in sway. For the OATRC data all movements in sway as well as wind forces have been neglected.

This simulation compromises a managed ice field, hence the results were checked against

	Croasdale	Modified Croasdale
Angle of ice accumulation (β)	15°	30°
Ambient ice pressure (p_a)	4 kPa	4 kPa
ice field cohesion (p_c)	2 kPa	$0.3p_a$

Table 10.1: Coefficients used in Croasdale resistance models.

the empirical Croasdale model from Section 3.1. The values used in the Croasdale model were taken from Kjerstad et al. (2018) based upon recommendations from Palmer and Croasdale (2014), and can be found in Table 10.1. In addition a modified version of the Croasdale formula with values deduced from rough simulation measurements and SAMS configuration parameters, can be found in Table 10.1.

10.1.1 Ice Field

The ice field was created from an aerial photo taken earlier in the day some hours before the full-scale transit was conducted. The aerial photo covers only a small area of the total transit distance, but are representative for the ice conditions during the transit. The process of creating the ice field was conducted by ArcISO. The ice field can be seen in Figure A.1, and has a thickness of 1 m. Note that for the full-scale test the exact ice thickness was unknown, but one observation during the test suggested an average thickness of 0.7 m and a maximum thickness of 1.2 m, however an observation before the transit suggested as little as 0.2 m and after the transit as much as 2.5 m.

10.2 Results

Figure 10.2 shows the ice force in surge from SAMS compared to the surge ice force from the full-scale OATRC trial, where the ice force from SAMS were filtered according to Section 8.1. In addition to the ice force, the calculated mean ice force and predicted ice force from the two Croasdale models are given in Figure 10.2. The mean ice force calculated by SAMS was 572 kN, and the mean ice force in surge calculated from OATRC data over the same time period was 480 kN. The Croasdale model from the managed ice load in Section 3.1 gave a predicted resistance of 884 kN, while the Croasdale model with the modified parameters gave a predicted resistance of 543 kN. Figure D.5 gives the velocity from the same time period for both the SAMS simulation and the OATRC trial. Here the mean velocity of SAMS was 5.68 m/s, and the mean velocity of OATRC was 5.87 m/s. The thruster force in surge for the same time period are given in Figure D.6, here the mean thrust of SAMS was 1.08 MN and for OATRC

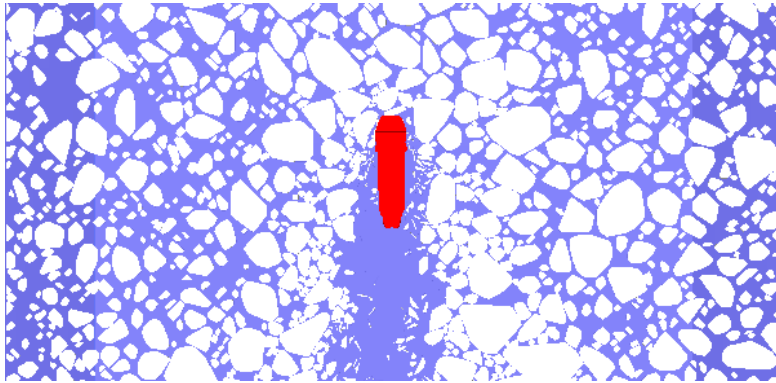


Figure 10.1: Image 110 s into the OATRC 2015 transit simulation.

it was calculated to 1.01 MN. A screenshot 110 s into the simulation can be seen in Figure 10.1.

Other results from the simulation yielded a maximum deviation in the sway position of just under 2 m and an oscillating sway velocity with amplitudes no higher than 0.2 m/s. The ice force in sway were varying fast and unpredictable, but had a mean value close to zero and the force were within ± 300 kN. The heading of the vessel was always within $\pm 1^\circ$ of the North direction while the ice force in heave was slowly varying with a mean of -700 kN, i.e. a force pushing the vessel upwards.

10.3 Discussion

For simplicity the path of Oden during the OATRC full-scale test was assumed to be in a straight line. This allowed the surge ice force from the simulator to be directly compared to the full-scale OATRC data, however this will not necessarily give correct results, e.g. if Oden was doing maneuvers that generated significant ice forces in sway. One other substantial assumption is that the wind velocity in the simulation is assumed to be zero, i.e. not exerting force on Oden and the surrounding ice field. This is however not true since the wind in the OATRC full-scale data was around 10 m/s coming from the starboard side of the vessel. Had this been taken into account in the simulation, the ice forces would likely increase due to added ice friction and increased ice concentration around the hull. As a result of this the simulation performed was an idealized test, and the results should be interpreted with this in mind.

10.3.1 Validation Against Empirical Data

From Figure 10.2 it can be seen that the mean results from SAMS are slightly higher, but in the right order of magnitude, compared to the ice forces from the OATRC trial.

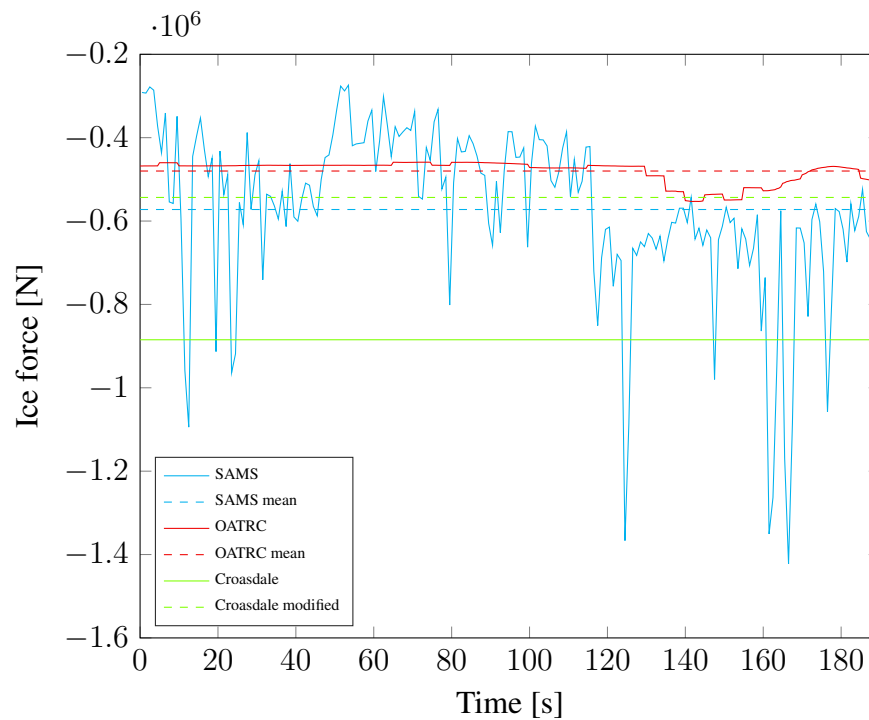


Figure 10.2: Predicted and simulated ice force in surge from OATRC simulation.

It should be noted that there is a high amount of uncertainty around this simulation as to which part of the OATRC data to compare against. The simulation data have been compared to a part of the OATRC data with relatively good accordance in velocity and somewhat in ice forces, suggesting that the time period is representative for the vessels movement, but not necessarily the concentration of the ice.

The fact that the ice thickness for the OATRC trial is most likely less than 1 m while that for the simulation is uniformly 1 m, could be the one obvious reason for the 19 % higher mean ice force from SAMS. The calculation method for thruster forces and thus ice forces could be another reason. Pitch and RPM of the propeller is not taken into account when the calculation of the thruster forces from the OATRC data was done, only engine power as the method described in Juva and Riska (2002) was used. This causes unreliable results as for instant, negative pitch on the propeller would still be interpreted as positive thrust by the method.

It can be seen that the ice force increases towards the end of the SAMS simulation, this could either be because of boundary effects with the wall or change in the ice field concentration and floe size. At the end of the simulation the vessel is still more than 100 m from the wall, so boundary effects are assumed to give neglectable contributions. Looking at the ice field in Figure A.1, a change in floe size and concentration is seen

and can explain the increase in ice resistance.

As can be seen from Figures 10.2 and D.5, the SAMS and OATRC results coincide reasonably well. The time period used in the comparison seems to be correct though the methods used to calculate the ice force from the OATRC data as well as the ice field configuration have a high degree of uncertainty as can be seen in Figure D.6, suggesting the need for additional tests and better methods for calculating thruster forces.

10.3.2 Controller Performance

It can be seen that the velocity controller is not able to hold a constant velocity in the ice field resulting in relatively large oscillations. Despite the oscillations the average velocity of the vessel is close to the setpoint of the controller, 5.68 m/s from the simulation compared to the average velocity from OATRC data of 5.69 m/s, however the trend is that the velocity is dropping towards the end. The drop in velocity can also be seen in the OATRC trials indicating that the time period for comparison could be in the correct region.

The oscillations seen in the velocity might be due to an aggressive tuning of the speed controller. The aggressive tuning can easily be seen in the thruster forces which are changing rapidly. It can also be seen that the thruster forces was saturated for a big part of the simulation, this is probably helping to stabilize the controller and lessen the oscillations. The rapid changes in the controller is unwanted as it wears out the thrusters and engines, thus a need for a slower controller is present.

The controller seems to handle the ice forces well. There is little movement in sway direction and the movements can in large be described with ice forces. The ice forces in heave direction most likely have an important part in the oscillation in sway velocity, when an ice floe is pushed under the vessel it will eventually go to one side before emerging at the side of the vessel, this produces forces in sway direction which can describe the oscillations in velocity. Though some oscillations are seen in the controller and the controller is a bit too responsive, the controller is deemed satisfactory for this simulation.

10.3.3 Validation Against Palmer and Croasdale (2014)

The comparison between SAMS and the Croasdale model show a 54 % higher ice force for the Croasdale model compared to SAMS. One reason for the high predicted ice force might be differences in the ice parameters used in the model, as all these were based upon recommendations from Palmer and Croasdale (2014). As of this a modified

Croasdale model is also shown, and the results are more in line with both the SAMS and OATRC mean results.

Figure 10.1 shows a screenshot 110 s into the transit simulation, and it can be seen that there are little sliding along the hull, indicating that a lower ice field cohesion is needed in the model. The ice-ice friction coefficient used in SAMS of 0.15 also indicates a lower ice field cohesion. The figure also shows small build up of floes in front of the vessel, indication a larger *ice build up angle*, β . These modifications to the Croasdale formula gives results more in line with SAMS and OATRC, indicating that for this particular ice field and Oden some tuning of the Croasdale parameters should be considered. Note however that there exist a large degree of uncertainty in the parameters used in the modified Croasdale model. In addition the velocity in the simulation was quite large, and this will introduce more uncertainty into the model parameters. For instance the large velocity leads to more breaking of floes, hence smaller build up and sliding along the hull.

10.3.4 Validation Against Kjerstad et al. (2018)

Figure 23 in Kjerstad et al. (2018) presents full-scale data from the same OATRC trial used for comparison in this thesis, obtaining a mean ice force of 822 kN. The ice forces from Kjerstad et al. (2018) was estimated using accelerometers, and give a substantially higher ice force than both SAMS and the calculated ice force from the OATRC trial using the method described in Juva and Riska (2002). The difference in ice forces from OATRC, using the Juva and Riska (2002) method and ice forces in Kjerstad et al. (2018), is mainly due to different ways of calculating thruster forces, where the mean thruster force from Kjerstad et al. (2018) is about 40 % higher than the results obtained by the Juva and Riska (2002) method. The results from Kjerstad et al. (2018) imply a propulsion efficiency of around 0.21 N/W which is around 70 % higher than what Keinonen et al. (1996) predicts for Oden at 5.7 m/s. This could mean that the thruster force should be lower than the one found in Kjerstad et al. (2018) resulting in lower estimated ice forces. However this could also mean that the calculation from OATRC data is giving a consistently low ice force, most likely the case is somewhere in between these two.

Notice that Lubbad et al. (2018) did a simulation with a newer version of SAMS and compared against the same OATRC 2015 trail data from 6.49AM to 7.05AM. The results from SAMS where in this case surprisingly close to the results from OATRC trial. This could indicate that the newer version of SAMS is more accurate and realistic, however it could also be a product of more tuning to get the results to match.

Transit Acceleration Feedback Simulation

The transit acceleration FB simulation in this chapter have been carried out in order to have a comparison for the reference controller without acceleration FB. The chapter will first present the setup of the simulation, including the setup of the reference setup without the acceleration FB controller, secondly the results from all the simulations will be presented. Lastly the results are discussed and compared to analyze which gives the best results.

11.1 Setup

To test the acceleration FB controller introduced in Section 7.4.2, a new simulation of the setup described in Section 10.1 was performed. The simulation was modified with a new velocity of 2 m/s and the acceleration FB gain was tuned to 0.6, 0.7 and 0.8 to give multiple data sets for comparison. In addition a reference simulation without the acceleration FB part of the controller was simulated in order to compare the performance of the different controllers. All other parameters was the same as the simulation in Section 10.1.

11.2 Results

Selected results from the transit simulations are shown in Figures 11.2 and 11.3 in addition to Figures D.7 and D.8. For these simulations the legend on each figure represents the acceleration FB gain used, where 0 is the reference controller used for comparison.

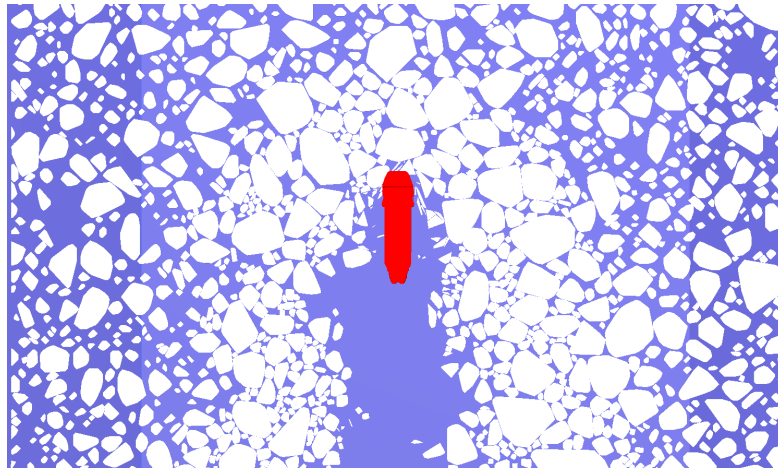


Figure 11.1: Image 360 s into the acceleration FB simulation with the reference controller.

Figure 11.2 shows the surge velocity for each simulation, Figure 11.3 shows the position in sway during the transit and Figure D.7 shows the thruster force in surge which have been filtered over 1 s. Figure D.8 shows the same thruster force as Figure D.7, but in its unfiltered state and for a smaller time period. In addition Figure 11.1 shows a screenshot of the simulation after 360 s, here with the reference controller, however for all intents and purposes the visualization of the different simulations looks the same.

The figures shown are the most relevant and interesting states for comparison, however it should be mentioned that the acceleration FB controller kept the heading within a maximum deviation from zero with 1.3° for every FB gain. Towards the end it seemed that the deviation started to increasingly oscillate, where every gain was in the same range as the reference controller. Results showed that the sway velocity had oscillations with period between 5 and 6 s and an amplitude of no more than 0.01 m/s in the acceleration FB controllers. Note that some smaller oscillations was also found in the reference controller. All gains showed decent performance in sway velocity and lay within the range of the reference controller with the maximum amplitude registered at 0.06 m/s. The combined thruster force in sway from the bow thrusters and azimuth thrusters show big oscillations, but lay in the same range as the reference controller.

11.3 Discussion

It can be seen from the results of the acceleration FB simulation that oscillations are present, this is true both for the reference controller and the acceleration FB controllers. It can though seem like the acceleration FB controllers are slightly more oscillating and more reactive, and this can be seen to also cause some instability towards the end. The acceleration FB controller also carries forward the fastest changing ice forces, which

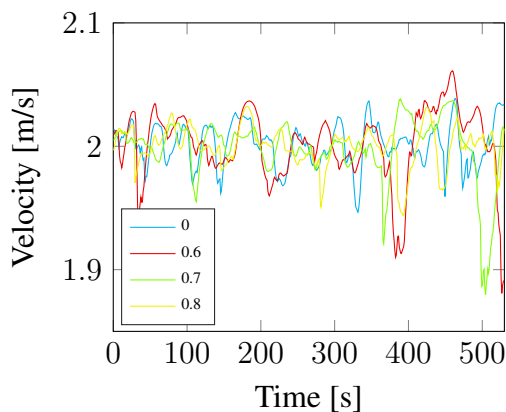


Figure 11.2: Surge velocities from acceleration FB simulations.

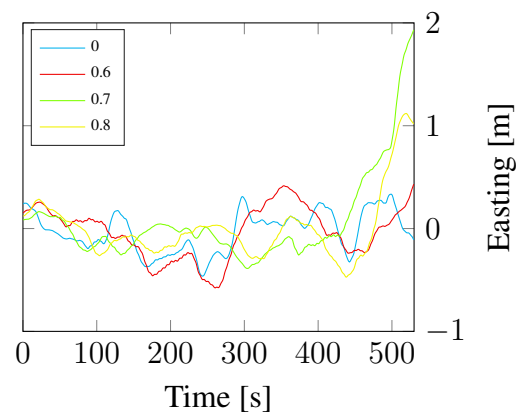


Figure 11.3: Easting position from acceleration FB simulations.

are unwanted, this can especially be seen in the unfiltered thruster force which are much more responsive and more noisy than the reference controller. This can be fixed by filtering the ice forces in a larger degree, however this will also introduce additional delay to the controller which are unwanted. Another solution could be filtering of the controller output, or a combination of the two solutions.

The simulation with acceleration FB does not seem to make much sense for this test and the results clearly show that it is hard to distinguish which gain gives the best performance. This indicates that the reference controller is fast enough to handle the ice forces on itself, one option could thus be to tune the controller slower and use the acceleration FB to handle the ice forces. This would then make the controller more stable and one could expect that the controller itself would give better results for environments where ice is not present. For this simulation the ice floes are also quite small, only giving slight ice forces with low peaks. However had the ice forces been larger, e.g. ice ridges, large ice floes, multi-year ice inclusions, one could expect better results from the acceleration FB controller.

Since the presented results are hard to analyze it is difficult to determine if the acceleration FB controller enhances the performance, and it can at some times seem like the acceleration FB controller is worse than the reference controller. It is also hard to determine a trend as to which acceleration FB gain performs best/worst, and the only conclusion that can be made is that the acceleration FB controller is not helping for this particular simulation and more simulations are needed to determine the performance of the acceleration FB controller. Based on the information from Section 4.2.1 that the acceleration FB adds a virtual mass to the vessel this suggests that such a controller are more suited for DP controllers than for maneuvering controllers.

THIS PAGE INTENTIONALLY LEFT BLANK

Dynamic Positioning and Ice Management Simulation

This chapter presents the two major simulations for this thesis, these includes two DP simulations in an ice field with and without IM support. The setup of the two simulations will be presented first, and the simulation without IM support will be presented secondly with results and a discussion. Lastly the results from the simulation with IM support will be presented and discussed.

12.1 Setup

Two DP simulations were set up in order to test the implemented controller, one IM strategy and the changes implemented into SAMS. The test was carried out in two separate simulations where the first was a lone DP vessel at the end of an ice field. In Simulink the reference model was excluded and replaced by a stationary reference position and a zero velocity. This means that the derivative part of the controller was trying to keep the velocity at zero even when out of position, and was chosen due to its simplicity, stabilizing properties and computational efficiency.

The second simulation was with the same DP vessel, but in addition an IM vessel tracking an arced racetrack path with its closest point approximately 450 m in front of the DP vessel. The modified version of Oden was used for both the vessels in the IM simulation, where the only difference between them being their starting position and their control system. The DP vessel utilized the same setup as in the DP simulation. While the IM vessel used the IM reference model and controller described in Sections 7.2.3 and 7.4.4 respectively. The IM vessel was set to hold a constant speed of 2.3 m/s and a

reduced speed of 1.4 m/s on the downstream turn in the West part of the path.

For the simulations the water current velocity was set to 0.3 m/s coming from North, i.e. to be coming directly toward the DP vessel at the open end of the simulation tank. Additionally the ice floes were initialized with the same velocity as the current so no velocity build up time was needed. Both the simulations were started as the DP vessel started to interact with the drifting ice field, and stopped approximately 100 m before the end of the ice field.

For the IM simulation removal of floes as described in Section 5.2.3 was performed, both the removal of floes outside of the tank and floes with a mass under 100 000 kg. This is approximately equivalent to an ice floe 10 m by 10 m in size and with a thickness of 1 m. For the DP simulation the same removal of floes was performed, but for ice floes with a mass under 1000 kg.

12.1.1 Ice Field

The ice field consists of three 1000 by 1500 m ice fields stitched together and created as described in Section 5.3. The ice field can be seen in Figure A.2 and was made from ice observations taken on Oden during OATRC 2015. In addition all ice floes smaller than 15 m in diameter were removed, this compromises around 10 to 25 % of the ice concentration observed around the vessel. At the far end of the ice field an additional ice field containing large ice floes was added, this was not part of the observed data, but was added to simulate ice pushing from behind. As of this the simulations were not conducted in this part of the ice field.

The time and date of the ice observations as well as the values used to create the ice fields can be found in Table 12.1. In the table $V_{n,min}$ and $V_{n,max}$ is the variable for controlling the roundness of the floes. Note that only the total ice floe-size distribution has been used. The distribution of first- and second-year ice as well as the snow depth has been ignored as the simulator do not take this into account. Spread in the thickness was not taken into account as only uniform ice thickness and properties are allowed. The ice field has been given a uniform thickness of 1 m.

12.2 Dynamic Positioning without Ice Management

12.2.1 Results

Selected results from the DP simulation without IM support are shown in Figure 12.2 and Figures D.9 to D.14. Figures 12.2 and D.9 show the surge and sway positions of the

Ice field	Oden data	Weibull IceMaker parameters			
		Concentration	λ	k	$\frac{V_{n,min}}{V_{n,max}}$
500	27.09.2015 17:52	0.85	13.5	0.83	$\frac{5}{8}$
1500	23.09.2015 05:19	0.85	6	0.75	$\frac{5}{8}$
2500	25.09.2015 14:16	0.88	4.6	0.58	$\frac{5}{8}$

Table 12.1: Data used to create the DP and IM ice field.

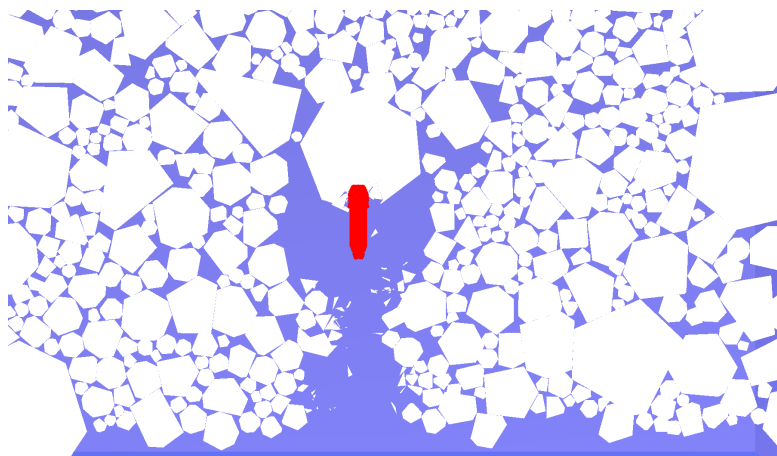


Figure 12.1: Image 1800 s into the DP simulation without IM support.

DP vessel respectively. Here it can be seen errors of up to ± 2 m in surge and ± 0.8 m in sway, however the mean position in surge and sway was within ± 1 cm. Figure D.10 shows the heading of the DP vessel and it can be seen that the heading error does not exceed 4.1° . Figures D.11 and D.12 show the filtered ice and thruster forces in surge respectively while Figures D.13 and D.14 show the filtered ice and thruster forces in sway. The ice forces give a minimum force of -400 kN and a mean force of -41 kN in surge, while a maximum of 98 kN and a mean force of 3.5 kN in sway are registered. Note that for this simulation the bow thrusters make up approximately two-thirds of the total sway force. A screenshot 1800 s into the simulation can be seen in Figure 12.1.

12.2.2 Discussion

It can be seen that during the entire test the vessel is not deviating more than 2.1 m from its reference position in surge and 1 m in sway. In the simulations, the vessel encounters ice floes up to 41×10^6 kg, i.e. more than 3 times the mass of the vessel, nevertheless the controller is fast and responsive enough to handle the added disturbance from the

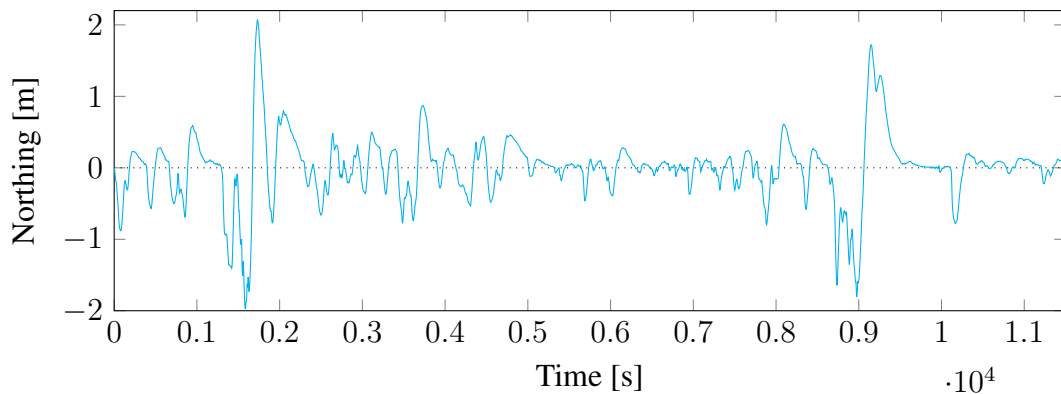


Figure 12.2: Northing position of Oden during DP simulation without IM support.

ice. The ice force in surge is quite low but with high peaks, despite the high peaks the controller manages to keep the vessel close to its setpoint. The ice forces would most likely increase significantly would the ice field have had a larger concentration. This was however not possible to accomplish as the ice fields are created as multiple single floes that cannot have any overlaps limiting the maximum concentration possible to generate. As for this simulation the concentration of floes are low enough such that the floes can in most cases drift around the vessel without exerting substantial force on the vessel.

Approximately 1500 s onwards compromises an area with constantly higher ice forces, this is due to a large ice floe being stuck around the bow of Oden. This increases the frontal area drastically and catches more ice floes in front of the vessel, an image 1800 s into the simulation of this can be seen in Figure 12.1. Note the increased channel width behind the vessel and the removal of ice floes at the far end of the figure. Even though the vessel is able to handle this situation and stays within a reasonable distance from its desired position, the situation should be avoided with IM support as the situation can easily escalate to a point where the vessel is stuck in the ice itself. In a denser ice field the situation would most likely either push the vessel off its setpoint or enclose it completely.

Another result of interest is the heading seen in Figure D.10, it can be seen that all angles are within 4.1° . In the simulation it is known that all ice floes and the water current comes straight from North, making a low heading error to the incoming floes ideal. However in more realistic situations the ice drift direction might change rapidly and ice drift and water current direction might not coincide. Additional parameters like wind, which could make up a large contribution if hitting from the side of the vessel, could together with worse ice conditions reduce the vessels ability to remain in DP substantially.

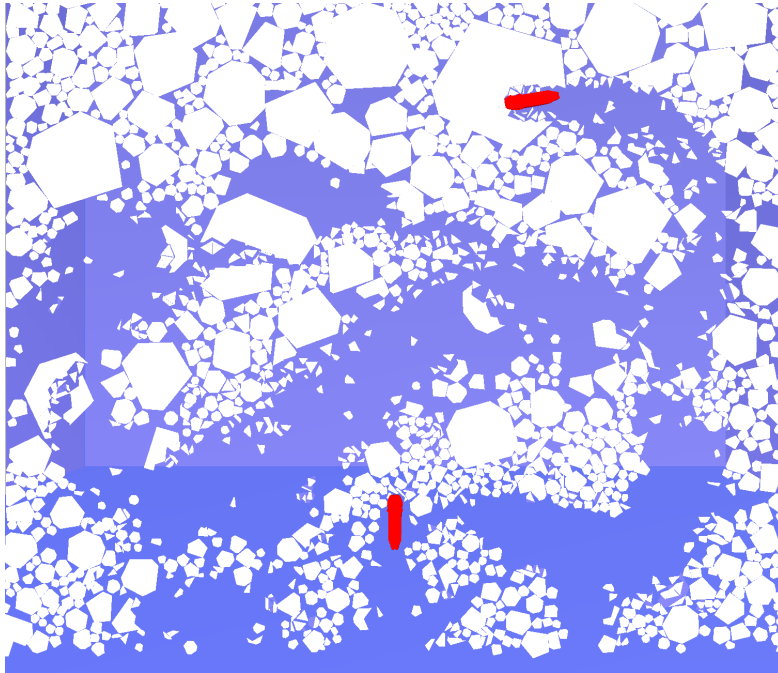


Figure 12.3: Image 8310 s into the DP and IM simulation, here the lower vessel is the DP vessel and the upper vessel is the IM vessel.

The controller performance is deemed as sufficient with the controller being responsive enough to handle even the biggest floes. However the simulation is unrealistic as no DP operations would operate in moderate ice conditions without IM support or some sort of mooring system. In addition the conditions are ideal, the ice concentration is sufficiently low with weak and reasonably thick ice, making the operation easier than what one could often expect in Arctic regions.

12.3 Dynamic Positioning with Ice Management

12.3.1 Results

Selected results from the DP simulation with IM support are shown in Figures 12.4 to 12.6 and Figures D.15 to D.25. The results are presented in Sections 12.3.1.1 and 12.3.1.2, where the results for the DP vessel are presented in the first section and the IM vessel in the second. A screenshot of the simulation after 8310 s, showing both the DP and IM vessels, can be seen in Figure 12.3.

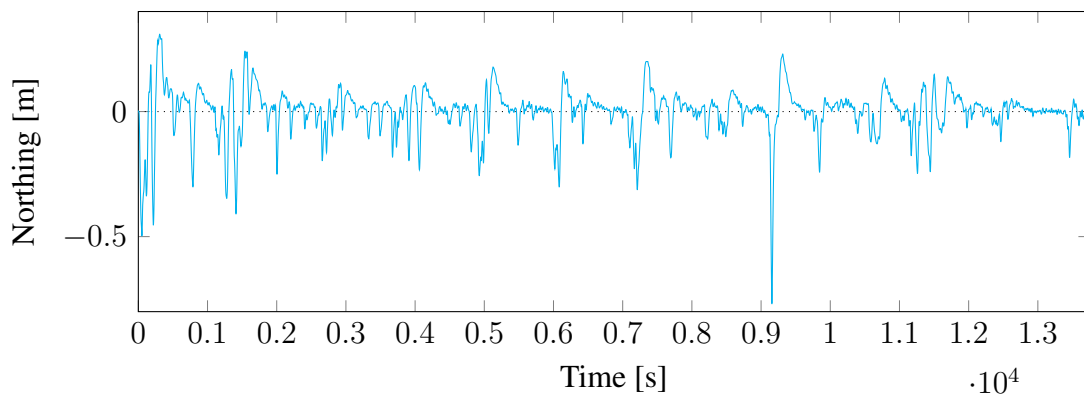


Figure 12.4: Northing position of Oden during DP simulation with IM support.

12.3.1.1 Dynamic Positioning Vessel

Figures 12.4 and D.15 presents the surge and sway position of the DP vessel with IM support with maximum errors of -0.77 m in surge, 0.48 m in sway and a mean position within 0.1 cm in both surge and sway. Figure D.16 shows the heading of the DP vessel where a maximum deviation of 2.2° can be seen. Figures D.17 and D.18 show the filtered ice and thruster forces in surge respectively while Figures D.19 and D.20 show the filtered ice and thruster forces in sway respectively. The minimum ice force in surge was -95 kN with a mean force of -5 kN, and in sway a maximum force of 113 kN, minimum of -75 kN and a mean force of 608 N was registered.

12.3.1.2 Ice Management Vessel

The IM vessel was set up to follow the arced racetrack path as described in Section 7.2.3. During the IM simulation the IM vessel managed to make 11 rounds of the arced racetrack path and the full path of the vessel are shown in Figure 12.5. In the figure, the outline of the arched racetrack is shown with dotted lines as a reference. Figure 12.6 show the surge velocity, this have a mean of 2.34 m/s with a maximum of 4.1 m/s and a minimum of 0.1 m/s. Figure D.21 presents the sway velocity, this has a mean of 0.4 m/s and a maximum and minimum of 2.4 m/s and -0.8 m/s respectively. The crab angle shown in Figure D.22 have been calculated based on the relative velocity, i.e. not the absolute velocity shown in the other figures, where the three values going outside the figure are deemed as not of interest due to low velocity at the time. The crab angle has a mean value of 9.8° , a maximum peak of 48° and a minimum peak of -23° , note that this is not including the peak values where the velocity is close to zero.

Figure D.24 shows the filtered thruster force in surge, the filtering was done with a mean filtering over 1 s. The thruster force has a peak of 1.2 MN, i.e. the maximum force

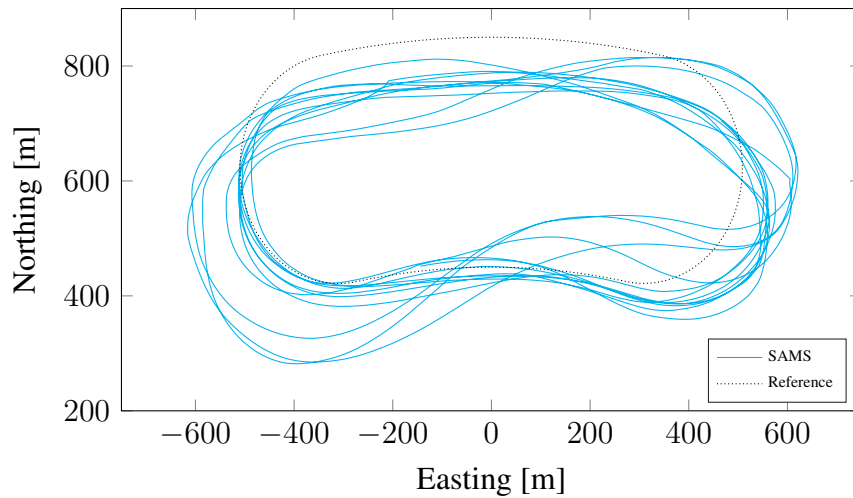


Figure 12.5: Path of the IM vessel during DP simulation with IM support.

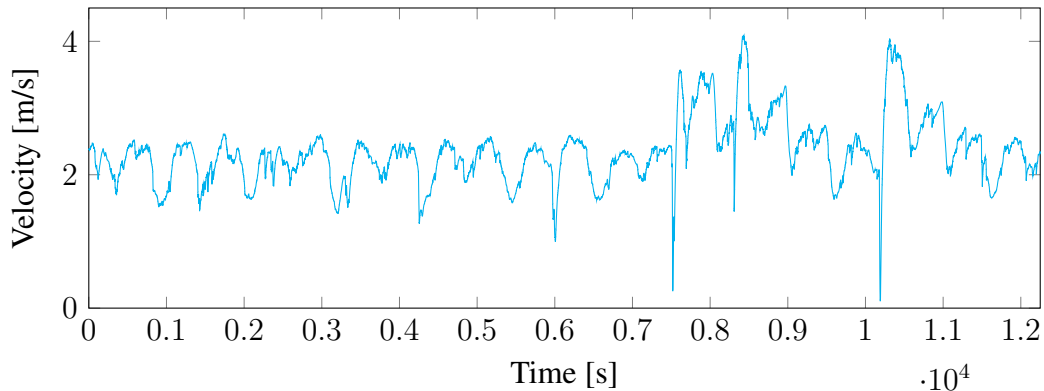


Figure 12.6: Surge velocity of the IM vessel during DP simulation with IM support.

the vessel can provide, while the mean force is 400 kN. The ice force in surge seen in Figure D.23 and has a peak of -4.75 MN and a mean force of -186 kN. Figure D.25 shows the filtered ice force in sway, this has a mean force of -95 kN and maximum peaks of -7.87 MN and 1.54 MN.

12.3.2 Discussion

The DP simulation with IM support are discussed in the following section. The discussion is divided into two parts, one involving the DP vessel and one involving the IM vessel. The IM vessel follows a predefined arched racetrack path which is common in the industry, the path however is quite close to the DP vessel and only a first stage IM operation is simulated. The IM vessel is creating a channel in the ice field, disrupting the drift of the ice floes and hence reducing their mean velocity. As a result of this the second DP simulation with IM support was simulated over a larger time period and the

results on the two DP vessels cannot directly be compared based on their timestamp in the figures.

12.3.2.1 Dynamic Positioning Vessel

Comparing the results from the DP vessel between the two simulations with and without IM support, it is easy to see that the IM support greatly improves the performance of the DP vessel. The maximum error in position are reduced by 36 % in surge and the mean position is one-tenth compared to the DP simulation without IM support. This indicates that the IM vessel manages to clear the area in front of the DP vessel, there are however seen some larger ice floes hitting the DP vessel in the simulation, this is mostly due to the fact that only a one stage IM support is carried out.

The performance improvement created as a result of the IM support is clearly seen in the surge ice force on the DP vessel. The mean ice force in surge registered on the DP vessel was reduced to one-eighth, and the maximum ice force was reduced to one-forth of its value from the DP simulation without IM support. This is credited to the IM vessel as it manages to create a channel in the ice field in front of the DP vessel with large areas of open water, resulting in large parts of the simulation with no ice forces, as can be seen in the figures, especially in Figure 12.3 where the channel can clearly be seen. Notice that the ice concentration is not only reduced in front of the DP vessel, but also on the sides, allowing the ice floes to drift around the DP vessel. It should also be said that a lot of the smaller ice floes has been removed by the IM vessel due to low mass after breaking. Would the ice floes not have been removed like this it would most likely not have any effect on the maximum ice force, but would probably give a larger mean and a more stable ice force. The removal of ice floes was however necessary in order to limit the total number of ice floes in the simulation.

From Figure 12.3 the close proximity of the two vessels during this simulation can also be seen, this is not ideal as real IM operations usually have a bigger safety distance to the DP vessel. However a larger safety distance would result in a larger ice field with an impractical number of floes making the simulation time significantly longer. As a result of the small safety distance the IM vessel with its racetrack path, interacted with the ice floes in the Southwest corner in such a way that the ice floes are pushed into the DP vessel. This is clearly seen in the ice force in sway on the DP vessel as it is oscillating around zero and the fact that the positive forces are recurring with a certain time period. Note that the maximum ice force in sway on the DP vessel are increased by 13 % compared to the DP vessel without IM support. Though this is only a slight increase, it would most likely have been bigger had the removal of floes not been present or put at a lower value. Note that this increase in sway ice force is not expected and is a

direct result of the close proximity between the vessels. This then shows the importance of planning the IM support, not only in a safe distance from the vessel, but also at a distance that reduces interactions between the vessels.

The thruster forces on the DP vessel with IM support can be seen to be more stable compared to the DP vessel without IM support, this is expected as the incoming ice field will have a lower concentration due to the channel created by the IM vessel. There are however seen some noise on the thruster forces in both simulations indicating that the controller is a bit too reactive. This is also seen in the position and heading as any errors are followed up by an overshoot creating an error in the opposite direction. The controller was tuned to be aggressive due to the ice, but this can indicate a too aggressive controller. Despite this the performance is considered to be well within the expectations for this thesis. The noise can also indicate that the LP filter is not able to filter out the added noise, this could be solved by tuning the LP filter more or by using more advanced filters such as a cascaded LP and notch filter. However, this would also introduce more unwanted delay in the system.

All the results point to the fact that the IM support greatly improves the DP capability of the DP vessel, this was as expected with the ice field used in the simulation. Limitations on the number of floes the simulator could handle, while still have reasonable simulation times limited the safety distance possible to use, hence some floes was unrealistically pushed towards the DP vessel. This indicates that care should be taken when choosing the path of the IM vessel to avoid this.

12.3.2.2 Ice Management Vessel

The path the IM vessel was supposed to follow was the arched racetrack described in Section 7.2.3, and it can be seen that a decent tracking is achieved. The general trend is that it overshoots a bit in the Southeast corner of the path and is not able to track the North part of the path. This is however acceptable as the reference model is most concerned with the heading of the IM vessel and the fact that it does not reach the North part of the path is of no particular concern. It can also be seen that on three of the laps it overshoots quite drastically in the West side of the path, this is due to the IM vessel hitting/getting stuck in relatively large ice floes in the North part of the path. Following the path it can be seen that all of the overshoots happen in the last half of the simulation as well as two happening straight after one another. This suggests that the controller is not able to stabilize itself in a large enough degree. At the same time it can suggest that the last part of the ice field proves to be too difficult to handle for the controller.

The ice forces on the vessel can be seen to be large, this is especially true for the sway

ice force. It can also be seen that the largest sway ice forces occur a few minutes after the large surge ice forces. The large sway ice forces also coincide with the overshoot of the Southwest part of the track indicating that they are a direct consequence of the larger floe in the last part of the ice field. The sway ice force is in these cases helping to minimize the overshoot as they will lower the crab angle in the turn. Though the vessel is designed to handle large ice forces the design often looks at ice forces from the bow or aft part of the vessel, large point loads from the sides might therefore cause damage to the hull of the vessel. This could in this instance be reduced by either increasing the turning radius or reduce the velocity. Reducing the velocity can however cause a problem with the vessel getting stuck in large ice floes.

It can be seen from the results that the velocity controller has some overshoot. In the simulation this makes it easier to break the floes and avoid being stuck, however in reality this would put extra stress on both the vessel and the crew. The ice forces from the increased velocity causes accelerations of more than 1 m/s^2 which for the small periods they are applied causes vibrations that put stress on both the crew as well as the vessel and its equipment. As of such the velocity controller should be tuned to be more damped to avoid the large overshoots.

Throughout the simulation it can be seen from Figure D.22 that the crab angle is for the most part positive. This is as expected as the vessel is for the most part conducting a counter clockwise turn, however the crab angle also causes the vessel to drift of path which can be seen especially in the Southeast part of the path. The crab angle is in some way taken into account in the long lookahead distance in the IM reference model, however this does not take difference in crab angle in the different part of the path or at different velocities into account. The current drift is neither taken into account and this should be calculated to get a crab angle from the current velocity. This could then be put into the controller to achieve better tracking. These two improvements would decrease the need for a long lookahead distance and get the vessel closer to its intended position, especially in the North part of the path.

As can be seen in Figure 12.3 the channel behind the IM vessel is quite wide and empty. This is mostly due to the removal of floes, but also partly the ice concentration and velocity of the vessel. The wide channels make the task easier as the ice floes are pushed to the sides more easily than what would be the case had the ice field had a higher ice concentration. The removal of ice floes also has a direct impact that they do not create sliding friction on the vessel while moving over the side and under the vessel. This likely reduce the overall ice resistance quite a lot and make the IM operations easier and less realistic than what would have been the case without the removal. However the ice floe removal is necessary as the introduction of small floes creates additional computations.

As the simulator needs to not only calculate the forces the ice floes has on the vessel, but also the forces they exert on each other and their interactions, the computational time increases quadratically with the amount of ice floes. Earlier tests showed that just two rounds with the IM vessel in the specified arched racetrack can double the amount of floes present if the minimum size of floes are set to 1000 kg instead of 100 000 kg. This would mean that the simulation carried out in this thesis would have to run for at least a few days to complete, making it highly impractical, if not also impossible.

It should also be mentioned that normal IM operations are not carried out by control systems, but by means of manual control. This is in part due to the unpredictable nature of ice which might call for changes in the path midway to either avoid or hit ice floes that might be a threat to the stationary vessel or structure. Manual control also makes it easier to conduct ramming, i.e. hitting ice floes or ice ridges multiple times, these maneuvers and when to do them can be difficult to calculate for a control system, especially since there often exists a lot of unknowns and even the manual control often are conducted more on instinct than after a specific procedure.

THIS PAGE INTENTIONALLY LEFT BLANK

Discussion of Simulator Improvements and Modifications

This chapter will first discuss some of the key elements regarding the implementations and modifications to the SAMS code and TCP interface made for this thesis. Lastly a discussion on the manual control application is made regarding its possible applications.

13.1 SAMS and the TCP Interface

One big consideration with the implementations carried out in SAMS for this thesis is that SAMS is under constant development, hence the implementations needed to be well documented in order for ArcISO to easily adapt them into their main version of SAMS. For this reason it was natural to implement the TCP interface in a separate library so it easily could be integrated into SAMS. However other parts of the implementations performed for this thesis, such as adding multiple vessels, could not be performed as a separate library because of how closely linked it is to the SAMS source code. These modifications were then carried out by trying to modify as little as possible in the main files of the SAMS source code and creating new versions of existing files needed, e.g. the *RHAS.itconfig* file, where the additional vessel data is specified. This resulted in having multiple files which have a similar function resulting in an unnecessary amount of files in the source code. The implementations was however performed in this manner at ArcISOs request since the code is under constant development, and files could be modified by ArcISO after the work on this thesis are concluded.

13.1.1 Performance Enhancements

The performance of the simulator has throughout the thesis been a problem, as of such the TCP interface and additional improvements in SAMS have been carried out in a way that minimizes the extra computations needed. One example is if there is a long distance¹ between the external software and SAMS on the network, or the external software needs prolonged time to compute results. In this case the simulation performance will reduce drastically. This have been tried solved by both unlinking the frequency of the simulator and external program, and computing the forces from the external program in parallel with the computations in SAMS. Unlinking the frequency from the simulator and the external software proved easy to accomplish, but as the control system in the external software in most cases need quite a high update rate and SAMS cannot simulate at a lower frequency than the external software, this solution proved to only add additional failure handling, complexity and computations. The solution with running in parallel though showed promising results. However as the rest of the simulator is sequential, the increase in performance was minimal and in a large degree gone because of additional failure handling. However it showed that the possibility of a simulator utilizing parallel processing and multiple cores can be possible. It should be noted that one of the main reason that this has not been performed by ArcISO is that this would make the simulator random, i.e. it would not be possible to recreate results.

Instead of trying to improve the computation efficiency the main performance enhancements have been on reducing the computational need. This is the case for the removal of floes when they drift out of a designated area and the removal of floes under a certain mass. Especially the first one can be utilized with no extra assumptions or introduction of additional uncertainties, if used correctly. This means that the removal of floes needs to be performed at a far enough distance from the vessels such that no effect from the ice floes interacts with the vessel(s). However this distance might not always be easy to determine and the method first shows its potential when the ice field is large such that a large part of the ice field can be removed during a simulation. This is the case in for instance the DP simulation with and without IM support.

The other case is the removal of floes once they are broken into smaller floes. The main problem with this method is that it removes ice that in reality should still be there, thus reducing the total concentration. It should be noted that even in the original SAMS code removal of small floes are carried out, this is necessarily should the calculation be possible for long time periods as small ice floes will drastically increase the com-

¹Refers to the total distance between datapoints (nodes) that the data-packets goes through on the network during the transmission. The bandwidth, latency and utilization of the network also affects the transmission time.

putations while not increasing the overall ice force by much. However finding a good value for this minimum floe size might be challenging as the need for efficiency and fast calculations need to be compromised with accuracy. This is one of the main reason the statistical output was implemented into SAMS in this thesis, namely to get the size of the removed ice floes. This makes post processing possible, speeding up the simulation as well as give good qualitative results for comparison of different IM strategies.

One other possibility to enhance the simulation performance is to create an area around the vessel in which the calculations on the ice floes are the same as in SAMS, while ice floes outside this area are regarded as *sleeping*. This means that for the sleeping floes no ice-ice calculations would need to be performed, thus reducing the calculations needed for the total ice field. When an ice floe drifts into the specified area around the vessel, it will need to be treated as an *awake* ice floe and ice-ice calculations would then be performed again. The challenge for this implementation is to determine the area for where the ice floes are regarded as sleeping, if this is chosen to small the ambient ice pressure around the vessel may not be correct, and if this is chosen to large the computational efficiency drops.

13.1.2 Vessel Interaction

The implementations completed in SAMS to allow for multiple vessels was achieved by modifying the original SAMS source code to allow for the additional initialization and handling of multiple vessels. The modifications did however not include collision detection and structure interactions between two vessels. This means that if a collision were to happen it will prompt an error message and SAMS will shut down. All simulations in this thesis have been carried out with the goal of no interactions, as of so a collision would make the simulation invalid. Simulations involving vessel contact, like docking or transfer of equipment or crew between vessels are seen as outside the scope of this thesis and therefore not considered.

13.1.3 Hydrodynamics

The possibility to compute the hydrodynamic forces in the external software removes the effect of the propeller wake between different vessels, however as discussed earlier, vessels will in most cases be far apart thus making this wake effect minimal. Note that even though the propeller wake do not have any effect between vessels, the ice floe interactions they create and forces they exert on ice floes are still active in the simulator. The main advantages of the external hydrodynamic calculation is better control and the ability to simulate more complex environments and add additional parameters to

the model. For instance wave forces or complex current environments can easily be added in an external software, while it is not possible to do in the simulator without direct access to the source code. In addition wind forces on the vessel can be calculated without the need to model the complete structure, as it needs to be if the wind forces in the simulator should be correctly applied.

13.1.4 Possibilities with TCP Interface

One of the major tasks for this thesis has been to open up the possibility to connect external software to SAMS, and thereby the possibility to implement a control system controlling a vessel in SAMS. This has been accomplished successfully where Simulink have been the external software used in this thesis. The use of Simulink opened the possibility to add a hydrodynamic model, hence calculate all hydrodynamic forces outside of SAMS. This can be pushed even further and the TCP interface created makes it possible for other simulators to connect to SAMS making it possible to simulate other parameters than what SAMS was originally intended to. For instance the vessel-ice interaction can be expanded to calculate deformations in the vessel hull using a suitable external software.

13.2 Manual Control

The manual control application described in Section 7.7 was created more as a proof of concept and to showcase the possibilities of the TCP interface implemented into SAMS. As this application utilizes rapidJSON as a serialization format the efficiency is reduced compared to sending raw data. However this provides great flexibility and the ability to use the same structure for multiple vessels.

The application is currently set up to only control the modified version of Oden used in this thesis, and in order to make this application more versatile it should be able to dynamically change the web application. This would make it possible to control any vessel with any thruster configuration, in addition the possibility to control multiple vessels should be implemented. This could be done either with multiple tabs in one such application or several applications each controlling one vessel. However the biggest improvement needed for this to be a useful solution is to speed up the simulation time, the easiest way to do it would be to use a more efficient serialization format as well as remove the proxy server.

Tests have shown that it is possible to set up the web application as a website which smartphones, tablets or computers can connect to and control a vessel in SAMS. The

main problem with the web application is the quantification of where the vessel is located in the simulation tank, since only numbers for the position and velocity are shown in the web application. Another problem is that there does not exist any way to limit the update frequency of the simulator, for manual control one would want a fixed time step, i.e. real-time, two times real-time etc. Before this is accomplished and the web application is set up to automatically configure and open both the simulator and the proxy server on startup, the web application has little practical use. However the application could with a few improvements be released together with SAMS as a separate module used for e.g. training or onboard the bridge of a real expedition vessel in the Arctic as a tool for the operator.

THIS PAGE INTENTIONALLY LEFT BLANK

Conclusion

The improvements and modifications to SAMS presented in this thesis have been successfully implemented where the improvements include, among others, the added possibility for several vessels, the removal of ice floes and the ability for external software to connect to SAMS. The TCP interface implemented shows that it is possible for external software to connect to SAMS and control one or several vessels in a reliable manner. It was shown that the removal of ice floes can greatly improve the simulation time, with the implementation of removing ice floes outside of a predefined scope and removal of ice floes under a certain mass. The removal of ice floes should however be used with caution with regards to both accuracy and simulation performance.

It has been shown that the connection to SAMS using the TCP interface can be achieved with different external software, making it a highly flexible solution. The manual control proved to be a good proof of concept, even though not used in the simulations in this thesis. With the use of Simulink the possibility to calculate hydrodynamic forces outside of SAMS is demonstrated, and can easily be extended to calculate other forces as well.

The control systems have shown that they are able to perform on a good level even in harsh ice conditions. Both the DP and IM control system performs better than expected and shows that the simulation of long and complex operations can be achieved in SAMS if the right precautions are taken. The acceleration FB controller however showed to be of little use, partly due to a responsive reference controller and partly due to relatively easy and constant ice conditions during the simulation. The simulator turned out to give reasonable results in both level ice and managed ice compared to empirical models and full-scale data, however the many uncertainties in ice parameters made it difficult to tune the simulator correctly to recreate full-scale trial data.

14.1 Further Work

There are still some work and improvements that can be implemented based on the discussion and conclusion. This can mainly be divided into three parts where the first part involves SAMS, the second part the TCP interface, and the last part involves the controllers.

Although every objective in the scope regarding the improvements and additional features to be added in SAMS was completed in a satisfactory manner, there are still some limitations found and they are:

- Multithreading for increased computational speed.
- Collision handler to handle collisions and contacts between vessels.
- Non uniform ice fields with different thicknesses as well as ice parameters, allowing for simulations of ice ridges and multi-year ice inclusion.
- Implementation of a zone around the vessel where every ice floe inside the zone are regarded as awake, and every ice floe outside the zone is regarded as sleeping.

The TCP interface have proven to be able to do more than the scope in this thesis, however some improvements could be:

- Utilization of asynchronous/parallel transmission of data.
- Data validity checks including the ability for the external software to decide if hydrodynamic forces should be calculated in SAMS at startup.

The controllers implemented for this thesis can be divided into two parts, where one part involves the manual control application and the second part involves the control systems implemented in Simulink. The manual control application here refers to the web application and the proxy server as illustrated in Figure 7.10, and the further work needed can be described as:

- Remove the need for a proxy server.
- Redesign of the GUI to make the layout dynamic with regards to the thruster controllers displayed.
- More efficient serialization format for improved performance.
- Support for multiple vessels.

All simulations carried out in Simulink was performed with simple controllers, and more complex controllers together with a more advanced hydrodynamics model are re-

quired if more advanced simulations are to be performed. When discussing the possible improvements to the controller this includes the observer, reference model and thruster allocation, and some of the improvements could be:

- Controller:
 - Hybrid control for both ice and open water maneuvering.
- Observer:
 - Signal fusion for several input signals.
 - Filtering of signals in case a loss of signal.
- Reference model:
 - Gain scheduling for different input amplitudes.
 - Inclusion of water current in IM reference model.
- Thruster allocation:
 - More advanced thruster dynamics.
 - Inclusion of rudder instead of azimuth thrusters.
- Hydrodynamics:
 - Tune damping and fluid memory effects to better coincide with trial data.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

- Çengel, Y., Cimbala, J., 2013. Fluid Mechanics: Fundamentals and Applications. Cengel series in engineering thermal-fluid sciences. McGraw-Hill Education.
- Choi, M., Chung, H., Yamaguchi, H., Nagakawa, K., 2015. Arctic sea route path planning based on an uncertain ice prediction model. *Cold Regions Science and Technology* 109, 61 – 69.
URL <https://doi.org/10.1016/j.coldregions.2014.10.001>
- Denk, F., 2016. Development of a Dynamic Positioning Toolbox for the Numerical Ice Tank Simulator. Tech. rep., Technische Universität München.
- Derradji-Aouat, A., jan 2010. Critical roles of constitutive laws and numerical models in the design and development of Arctic offshore installations. International Conference and Exhibition on Performance of Ships and Structures in Ice 2010, (ICETECH).
- DNV, 2011. Rules for classification of ships Pt.6 Ch.7 - Dynamic Positioning Systems.
URL <https://rules.dnvgl.com/docs/pdf/DNV/ruleship/2012-01/ts607.pdf>
- Du, J., Hu, X., Liu, H., Chen, C. L. P., nov 2015. Adaptive Robust Output Feedback Control for a Marine Dynamic Positioning System Based on a High-Gain Observer. *IEEE Transactions on Neural Networks and Learning Systems* 26 (11), 2775–2786.
URL <https://doi.org/10.1109/tnnls.2015.2396044>
- Egeland, O., 2002. Modeling and simulation for automatic control. Marine Cybernetics, Trondheim.
- Eik, K. J., 2010. Ice Management in Arctic Offshore Operations and Field Developments. Ph.D. thesis, Norwegian University of Science and Technology.
URL <http://hdl.handle.net/11250/231776>

-
- Erceg, S., Ehlers, S., 2017. Semi-empirical level ice resistance prediction methods. *Ship Technology Research* 64 (1), 1–14.
URL <https://doi.org/10.1080/09377255.2016.1277839>
- Fall, K. R., Stevens, W. R., 2012. *TCP/IP illustrated*. Addison-Wesley professional computing series *TCP/IP illustrated*. Addison Wesley.
- Faltinsen, O. M., 1990. *Sea loads on ships and offshore structures*. Cambridge ocean technology series. Cambridge University Press, Cambridge.
- faqs.org, 2014. *Raw IP Networking FAQ*. <http://www.faqs.org/faqs/internet/tcp-ip/raw-ip-faq/>, Accessed: 10.04.2018.
- Fernandes, D. d. A., 2015. *An output feedback motion control system for ROVs : guidance, navigation, and control*. Ph.D. thesis, Norwegian University of Science and Technology.
URL <http://hdl.handle.net/11250/299183>
- Fossen, T. I., apr 2011. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd.
URL <https://doi.org/10.1002/9781119994138>
- Fossen, T. I., Fjellstad, O.-E., jan 1995. Nonlinear modelling of marine vehicles in 6 degrees of freedom. *Mathematical Modelling of Systems* 1 (1), 17–27.
URL <https://doi.org/10.1080/13873959508837004>
- Fossen, T. I., Perez, T., 2004. *Marine Systems Simulator (MSS)*.
URL <http://www.marinecontrol.org>
- Fossen, T. I., Sagatun, S. I., 1991. Adaptive control of nonlinear underwater robotic systems. *Modeling, Identification and Control: A Norwegian Research Bulletin* 12 (2), 95–105.
URL <https://doi.org/10.4173/mic.1991.2.4>
- Fossen, T. I., Strand, J. P., 1999. Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel. *Automatica* 35 (1), 3 – 16.
URL [https://doi.org/10.1016/S0005-1098\(98\)00121-6](https://doi.org/10.1016/S0005-1098(98)00121-6)
- Girard, L., Weiss, J., Molines, J. M., Barnier, B., Bouillon, S., 2009. Evaluation of high-resolution sea ice models on the basis of statistical and scaling properties of Arctic sea ice drift and deformation. *Journal of Geophysical Research: Oceans* 114 (C8),

c08015.

URL <https://doi.org/10.1029/2008JC005182>

Glaessgen, E., Stargel, D., apr 2012. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics.

URL <https://doi.org/10.2514/6.2012-1818>

Grievess, M., Vickers, J., 2016. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Transdisciplinary Perspectives on Complex Systems. Springer International Publishing, pp. 85–113.

URL https://doi.org/10.1007/978-3-319-38756-7_4

Gürtner, A., Baardson, B. H. H., Kaasa, G.-O., Lundin, E., 2012. Aspects of Importance Related to Arctic DP Operations.

URL <https://doi.org/10.1115/OMAE2012-84226>

Hamilton, J. M., 2011. The Challenges of Deep-Water Arctic Development. International Society of Offshore and Polar Engineers 21.

URL <https://www.onepetro.org/download/journal-paper/ISOPE-11-21-4-241?id=journal-paper/ISOPE-11-21-4-241>

Haugen, J., Imstrand, L., Løset, S., Skjetne, R., jun 2011. Ice Observer System for Ice Management Operations. Proceedings of the International Offshore and Polar Engineering Conference.

URL https://www.researchgate.net/publication/237063038_Ice_Observer_System_for_Ice_Management_Operations

Hibler III, W. D., 1979. A Dynamic Thermodynamic Sea Ice Model. Journal of Physical Oceanography 9 (4), 815–846.

URL [https://doi.org/10.1175/1520-0485\(1979\)009<0815:ADTSIM>2.0.CO;2](https://doi.org/10.1175/1520-0485(1979)009<0815:ADTSIM>2.0.CO;2)

Holub, C., Matskevitch, D., Kokkinis, T., Shafrova, S., 2018. Near-field ice management tactics – Simulation and field testing. Cold Regions Science and Technology.

URL <https://doi.org/10.1016/j.coldregions.2018.02.003>

Hu, J., Zhou, L., 2016. Further study on level ice resistance and channel resistance for an icebreaking vessel. International Journal of Naval Architecture and Ocean Engi-

-
- neering 8 (2), 169 – 176.
URL <https://doi.org/10.1016/j.ijnaoe.2016.01.004>
- Hunke, E. C., Dukowicz, J. K., 1997. An Elastic–Viscous–Plastic Model for Sea Ice Dynamics. *Journal of Physical Oceanography* 27 (9), 1849–1867.
URL [https://doi.org/10.1175/1520-0485\(1997\)027<1849:AEVPMF>2.0.CO;2](https://doi.org/10.1175/1520-0485(1997)027<1849:AEVPMF>2.0.CO;2)
- IBM, 2013a. How sockets work. https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/howdosockets.htm, Accessed: 10.04.2018.
- IBM, 2013b. Socket characteristics. https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/characteristics.htm, Accessed: 10.04.2018.
- IBM, 2013c. Socket programming. https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/rzab6/rzab6soxoverview.htm, Accessed: 10.04.2018.
- Janßen, C. F., 2016. elbe. <https://www.tuhh.de/elbe/home.html>, accessed: 09.10.2017.
- Janßen, C. F., Mierke, D., Rung, T., 2017. On the development of an efficient numerical ice tank for the simulation of fluid-ship-rigid-ice interactions on graphics processing units. *Computers & Fluids* 155 (Supplement C), 22 – 32, iCMMES2015.
URL <https://doi.org/10.1016/j.compfluid.2017.05.006>
- Jayasiri, A., Nandan, A., Imtiaz, S., Spencer, D., Islam, S., Ahmed, S., dec 2016. An optimum control-based approach for Dynamic Positioning of vessels. In: 2016 IEEE 55th Conference on Decision and Control (CDC). IEEE, pp. 2796–2801.
URL <https://doi.org/10.1109/cdc.2016.7798685>
- Jenssen, N. A., Muddesitti, S., Phillips, D., Backstrom, K., 2009. DP In Ice Conditions. http://dynamic-positioning.com/proceedings/dp2009/arctic_jenssen.pdf, accessed: 12.10.2017.
- Juva, M., Riska, K., 2002. On the power requirement in the Finnish-Swedish ice class rules. Winter navigation Research Board, Res. Rpt 53.
- Keinonen, A., Browne, R. P. Revill, C., 1991. Icebreaker design synthesis : phase 2 : analysis of contemporary icebreaker performance. AKAC Inc. report for Transportation Development Centre, Transport Canada, TP 10923E.

-
- Keinonen, A., Browne, R. P. Revill, C., A, R., 1996. Icebreaker characteristics synthesis. AKAC Inc. report for Transportation Development Centre, Transport Canada, TP 12812E.
- Keinonen, A., Wells, H., Dunderdale, P., Pilkington, R., Miller, G., Brovin, A., 1999. ISOPE-I-00-102. International Society of Offshore and Polar Engineers, Ch. Dynamic Positioning Operation In Ice, Offshore Sakhalin May – June 1999, pp. 683–690.
- Kjerstad, Ø. K., Lu, W., Skjetne, R., Løset, S., mar 2018. A method for real-time estimation of full-scale global ice loads on floating structures. *Cold Regions Science and Technology*.
URL <https://doi.org/10.1016/j.coldregions.2018.03.012>
- Kjerstad, Ø. K., Skjetne, R., 2014. Modeling and Control for Dynamic Positioned Marine Vessels in Drifting Managed Sea Ice. *Modeling, Identification and Control* 35 (4), 249–262.
URL <https://doi.org/10.4173/mic.2014.4.3>
- Kjerstad, Ø. K., Skjetne, R., Jenssen, N. A., jan 2011. Disturbance rejection by acceleration feedforward: Application to dynamic positioning. *IFAC Proceedings Volumes* 44 (1), 2523–2528.
URL <https://doi.org/10.3182/20110828-6-it-1002.03454>
- Lee, S.-G., Zhao, T., Kim, G.-S., Park, K.-D., 2013. ISOPE-I-13-147. International Society of Offshore and Polar Engineers, Ch. Ice Resistance Test Simulation of Arctic Cargo Vessel Using FSI Analysis Technique, pp. 1162–1168.
- Leppäranta, M., 2005. *The Drift of Sea Ice*. Springer Berlin Heidelberg.
URL <https://doi.org/10.1007/b138386>
- Lindqvist, G., 1989. A straightforward method for calculation of ice resistance of ships. *Proceedings of the 10th International Conference on Port and Ocean Engineering Under Arctic Conditions*.
- Loe, Ø. A. G., 2008. *Collision Avoidance for Unmanned Surface Vehicles*. Ph.D. thesis, Norwegian University of Science and Technology.
URL <http://hdl.handle.net/11250/259696>
- Lubbad, R., Løset, S., 2011. A numerical model for real-time simulation of ship–ice interaction. *Cold Regions Science and Technology* 65 (2), 111 – 127.
URL <https://doi.org/10.1016/j.coldregions.2010.09.004>
-

-
- Lubbad, R., Løset, S., Lu, W., Tsarau, A., van den Berg, M., may 2018. An overview of the Oden Arctic Technology Research Cruise 2015 (OATRC2015) and numerical simulations performed with SAMS driven by data collected during the cruise. *Cold Regions Science and Technology*.
URL <https://doi.org/10.1016/j.coldregions.2018.04.006>
- Metrikin, I., 2014. A Software Framework for Simulating Stationkeeping of a Vessel in Discontinuous Ice. *Modeling, Identification and Control* 35 (4), 211–248.
URL <https://doi.org/10.4173/mic.2014.4.2>
- Metrikin, I., Borzov, A., Raed, L., Løset, S., 2012. Numerical Simulation of a Floater in a Broken-Ice Field: Part II — Comparative Study of Physics Engines. *ASME 2012 31st International Conference on Ocean, Offshore and Arctic Engineering* 6, 477–486.
URL <https://doi.org/10.1115/OMAE2012-83430>
- Metrikin, I., Kerkeni, S., Jochmann, P., Løset, S., jun 2015. Experimental and Numerical Investigation of Dynamic Positioning in Level Ice. *Journal of Offshore Mechanics and Arctic Engineering* 137 (3), 031501–031501–10.
URL <https://doi.org/10.1115/1.4030042>
- Metrikin, I., Løset, S., Jenssen, N. A., Kerkeni, S., 2013. Numerical Simulation of Dynamic Positioning in Ice. *Marine Technology Society Journal* 47 (2), 14–30.
URL <https://doi.org/10.4031/MTSJ.47.2.2>
- Moran, K., Backman, J., Farrell, J. W., 2006. Deepwater drilling in the Arctic Ocean's permanent sea ice. *Proceedings of the integrated Ocean drilling Program* MS 302-106 302.
URL <https://doi.org/10.2204/iodp.proc.302.106.2006>
- Nevel, D., 1961. Semi-infinite plate on elastic foundation. *CRREL Research Report*.
- Nguyen, D. T., Sørbø, A. H., Sørensen, A. J., 2009. MODELLING AND CONTROL FOR DYNAMIC POSITIONED VESSELS IN LEVEL ICE. *IFAC Proceedings Volumes* 42 (18), 229–236.
URL <https://doi.org/10.3182/20090916-3-br-3001.0006>
- Palmer, A., Croasdale, K. R., 2014. *Arctic Offshore Engineering*. World Scientific Publishing Co Pte Ltd, Singapore, United States.
URL <http://ebookcentral.proquest.com/lib/ntnu/detail.action?docID=1080968>
-

-
- Perez, T., Fossen, T. I., 2007. Kinematic Models for Manoeuvring and Seakeeping of Marine Vessels. *Modeling, Identification and Control: A Norwegian Research Bulletin* 28 (1), 19–30.
URL <https://doi.org/10.4173/mic.2007.1.3>
- Riska, K., 1998. Performance of merchant vessels in ice in the Baltic. Helsinki University of Technology, Ship Laboratory.
- Riska, K., 2010a. Design of ice breaking ships. *Cold Regions Science and Technology*.
URL <http://www.eolss.net>
- Riska, K., 2010b. Ship–Ice Interaction in Ship Design: Theory and Practice. *Cold Regions Science and Technology*.
URL <http://www.eolss.net>
- Rohlén, Å., 2009. Relationship Between Ice-Management and Station Keeping in Ice. *Dynamic Positioning Conference*.
URL http://dynamic-positioning.com/proceedings/dp2009/arctic_rohlen_pp.pdf
- Schreyer, H. L., Sulsky, D. L., Munday, L. B., Coon, M. D., Kwok, R., 2006. Elastic-decohesive constitutive model for sea ice. *Journal of Geophysical Research: Oceans* 111 (C11), c11S26.
URL <https://doi.org/10.1029/2005JC003334>
- Shannon, C., jan 1949. Communication in the Presence of Noise. *Proceedings of the IRE* 37 (1), 10–21.
URL <https://doi.org/10.1109/jrproc.1949.232969>
- Skjetne, R., Imsland, L., Løset, S., 2014. The Arctic DP Research Project: Effective Stationkeeping in Ice. *Modeling, Identification and Control: A Norwegian Research Bulletin* 35 (4), 191–210.
URL <https://doi.org/10.4173/mic.2014.4.1>
- Smogeli, Ø., 2017. Digital Twins at work in Maritime and Energy.
- SNAME, 1950. Nomenclature for Treating the Motion of a Submerged Body Through a Fluid: Report of the American Towing Tank Conference. Technical and research bulletin. Society of Naval Architects and Marine Engineers.
- Sørensen, A. J., 2011. A survey of dynamic positioning control systems. *Annual Reviews in Control* 35 (1), 123 – 136.
URL <https://doi.org/10.1016/j.arcontrol.2011.03.008>
-

-
- Sørensen, A. J., 2012. Marine Control Systems Propulsion and Motion Control of Ships and Ocean Structures Lecture Notes.
- Su, B., Skjetne, R., Berg, T. E., 2014. Numerical assessment of a double-acting offshore vessel's performance in level ice with experimental comparison. *Cold Regions Science and Technology* 106-107, 96 – 109.
URL <https://doi.org/10.1016/j.coldregions.2014.06.012>
- Tan, X., Riska, K., Moan, T., 2014. Effect of dynamic bending of level ice on ship's continuous-mode icebreaking. *Cold Regions Science and Technology* 106-107, 82 – 95.
URL <https://doi.org/10.1016/j.coldregions.2014.06.011>
- Valanto, P., Puntigliano, F. M. E., 1997. A numerical prediction of the icebreaking resistance of a ship in level ice: Final report. Technical report, Hamburg Ship Model Basin.
- Vance, G. P., 1980. Analysis of the performance of a 140-ft Great Lakes icebreaker USCGC Katmai Bay. U.S. Army Cold Regions Research and Engineering Laboratory, report No. 80-8.
URL <https://trid.trb.org/view/155590>
- Wang, J., Derradji-Aouat, A., nov 2010. Ship Performance in Broken Ice Floes - Preliminary Numerical Simulations. NRC Publications Archive (NPARC).
URL <https://doi.org/10.4224/17210723>
- Weiss, J., Schulson, E. M., Stern, H. L., 2007. Sea ice rheology from in-situ, satellite and laboratory observations: Fracture and friction. *Earth and Planetary Science Letters* 255 (1), 1 – 8.
URL <https://doi.org/10.1016/j.epsl.2006.11.033>
- Witkowska, A., aug 2013. Dynamic positioning system with vectorial backstepping controller. In: 2013 18th International Conference on Methods Models in Automation Robotics (MMAR). pp. 842–847.
URL <https://doi.org/10.1109/MMAR.2013.6670022>
- Wright, B., 1999. Evaluation of full-scale data for moored vessel stationkeeping in pack ice. Tech. rep., National Research Council Canada.
URL <https://doi.org/10.4224/12340932>
- Yulmetov, R., Lubbad, R., Løset, S., jan 2016. Planar multi-body model of iceberg free drift and towing in broken ice. *Cold Regions Science and Technology* 121, 154–166.
URL <https://doi.org/10.1016&j.coldregions.2015.08.011>

Zhang, Q., Zhang, X., Xu, W., Liu, A., Zhou, Z., Pham, D. T., 2017. Modeling of Digital Twin Workshop Based on Perception Data. In: Huang, Y., Wu, H., Liu, H., Yin, Z. (Eds.), *Intelligent Robotics and Applications*. Springer International Publishing, Cham, pp. 3–14.

URL https://doi.org/10.1007/978-3-319-65298-6_1

Ziegler, J. G., Nichols, N. B., 1993. Optimum Settings for Automatic Controllers. *Journal of Dynamic Systems, Measurement, and Control* 115 (2B), 220.

URL <https://doi.org/10.1115/1.2899060>

THIS PAGE INTENTIONALLY LEFT BLANK

Ice Fields

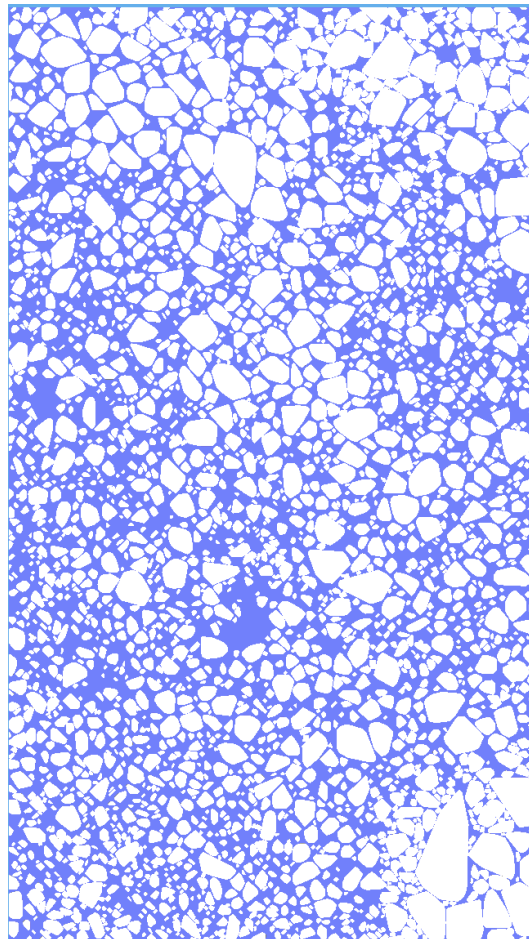


Figure A.1: Ice field used for OATRC 2015 transit and acceleration FB simulations.

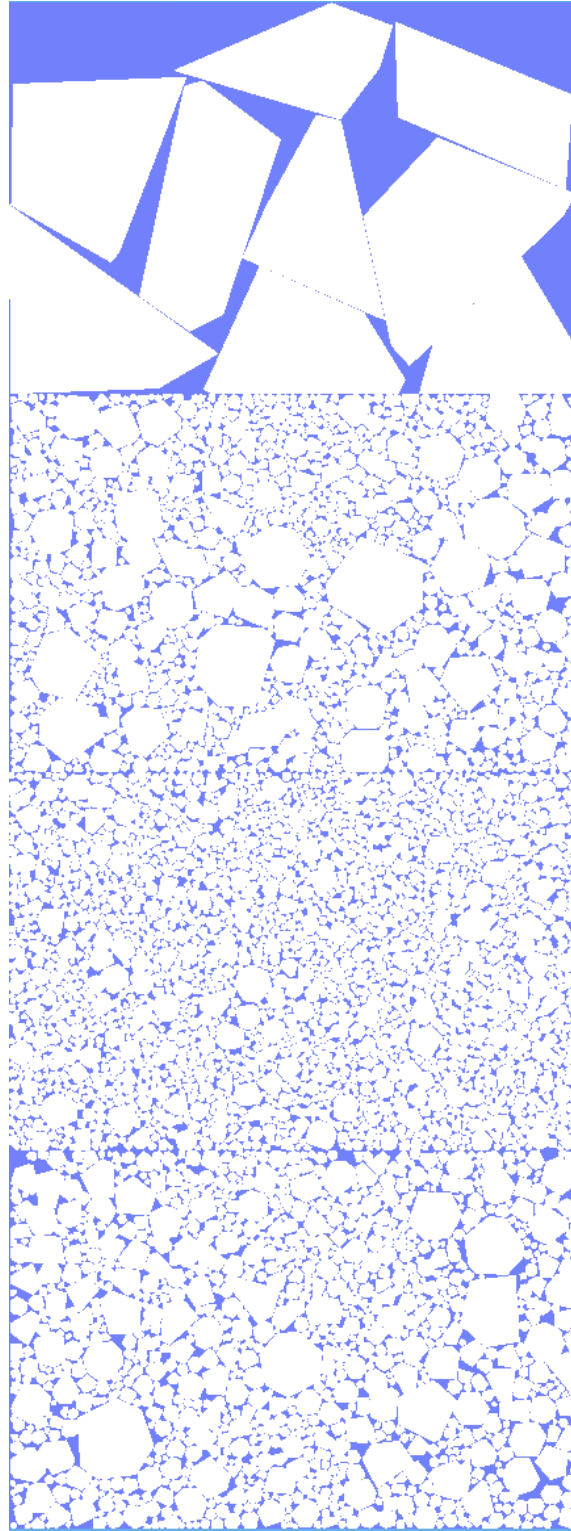


Figure A.2: Ice field used for DP and IM simulations.

Ice Resistance Models

B.1 Lindqvist (1989)

$$R_{ice} = (R_c + R_b) \left(1 + \frac{1.4V}{\sqrt{gh}} \right) + R_s \left(1 + \frac{9.4V}{\sqrt{gL}} \right),$$

$$R_c = F_v \frac{\tan \varphi + (\mu_{si} \cdot \cos \varphi / \cos \Psi)}{1 - (\mu_{si} \cdot \cos \varphi / \cos \Psi)} h^2,$$

$$R_b = \frac{27}{64} \sigma_f B \left(\frac{h^{1.5}}{\sqrt{\frac{E}{12(1-\nu^2)\rho_{water}g}}} \right) \left(\tan \varphi + \frac{\mu_{si} \cdot \cos \varphi}{\sin \alpha \cdot \cos \Psi} \right) \left(1 + \frac{1}{\cos \Psi} \right),$$

$$R_s = \Delta \rho g h B \left(T \frac{B+T}{B+2T} + \mu_{si} \left(0.7L - \frac{T}{\tan \varphi} - \frac{B}{4 \tan \alpha} + T \cos \varphi \cos \Psi \sqrt{\frac{1}{\sin^2 \varphi} + \frac{1}{\tan^2 \alpha}} \right) \right),$$

$$F_v = 0.5 \sigma_f h^2,$$

$$\Psi = \arctan \left(\frac{\tan \varphi}{\sin \alpha} \right).$$

Ship Parameters	V	Ship speed
	L	Ship length
	B	Ship beam
	T	Ship draft
	α	Waterline angle
	Ψ	Average bow flare angle at waterline
	φ	Stem angle
Ice parameters	h	Ice thickness
	σ_f	Flexural strength
	E	Young's modulus
	ν	Poisson's coefficient
	ρ_{ice}	Density of ice
	ρ_{water}	Density of water
	μ_{si}	Structure-ice friction coeff.

Table B.1: Table of parameters required for the Lindqvist resistance model.

B.2 Keinonen and Browne (1991); Keinonen et al. (1996)

$$R(V)_{total} = R(V)_{ow} + R(1 \text{ m/s})_{ice} + R(> 1 \text{ m/s})_{ice}$$

For ships with fully formed bows:

$$\begin{aligned}
 R(1 \text{ m/s})_{ice} &= 0.015 \cdot h_e^{1.5} \cdot C_s \cdot && \text{Ship size correction} \\
 & B^{0.7} \cdot L^{0.2} \cdot T^{0.1} \cdot && \text{Friction correction} \\
 & (1 - 0.0083(t + 30)) \cdot C_h \cdot && \text{Ice strength correction} \\
 & (0.63 + 0.00074\sigma_f) \cdot && \text{Bow form correction} \\
 & (1 + 0.0018(90 - \Psi)^{1.6}) \cdot (1 + 0.003(\beta - 5)^{1.5}), && \\
 R(> 1 \text{ m/s})_{ice} &= 0.009 \cdot (V_{increase} / \sqrt{gL}) \cdot && \text{Ship size correction} \\
 & B^{1.5} \cdot L^{0.5} \cdot h_e \cdot && \text{Friction correction} \\
 & (1 - 0.0083(t + 30)) \cdot C_h \cdot && \text{Bow form correction} \\
 & (1 + 0.0018(90 - \Psi)^{1.6}) \cdot (1 + 0.004(\beta - 5)^{1.5}). &&
 \end{aligned}$$

For ships with sharp chined shoulders:

$$\begin{aligned}
 R(1 \text{ m/s})_{ice} &= 0.08 + 0.017 \cdot h_e^{1.25} \cdot C_s \cdot \\
 &\quad B^{0.7} \cdot L^{0.2} \cdot T^{0.1} \quad \text{Ship size correction} \\
 &\quad (1 - 0.0083(t + 30)) \cdot C_h \quad \text{Friction correction} \\
 &\quad (0.63 + 0.00074\sigma_f) \cdot \quad \text{Ice strength correction} \\
 &\quad (1 + 0.0018(90 - \Psi)^{1.4}) \cdot (1 + 0.004(\beta - 5)^{1.5}), \quad \text{Bow form correction} \\
 R(> 1 \text{ m/s})_{ice} &= 0.009 \cdot (V_{increase}/\sqrt{gL}) \cdot \\
 &\quad B^{1.5} \cdot L^{0.5} \cdot h_e \quad \text{Ship size correction} \\
 &\quad (1 - 0.0083(t + 30)) \cdot C_h \quad \text{Friction correction} \\
 &\quad (1 + 0.0018(90 - \Psi)^{1.4}) \cdot (1 + 0.003(\beta - 5)^{1.5}). \quad \text{Bow form correction}
 \end{aligned}$$

Ship Parameters	V	Ship speed
	L	Ship length
	B	Ship beam
	T	Ship draft
	β	Average buttock angle at waterline
	Ψ	Average bow flare angle at waterline
Ice parameters	h	Ice thickness
	σ_f	Flexural strength
	ρ_{water}	Density of water
	h_s	Snow thickness
	t	Ice surface temperature
	C_h	Hull condition factor
	C_s	Salinity of water factor

Table B.2: Table of parameters required for the Keinonen resistance model.

B.3 Riska (1998)

$$R_{ice} = C_1 + C_2V,$$

$$C_1 = f_1 \frac{1}{2\frac{T}{B} + 1} BL_{par}h + (1 + 0.0021\varphi)(f_2Bh^2 + f_3L_{bow}h^2 + f_4BL_{bow}h),$$

$$C_2 = (1 + 0.0063\varphi)(g_1h^{1.5} + g_2Bh) + g_3h \left(1 + 1.2\frac{T}{B}\right) \frac{B^2}{\sqrt{L}}.$$

	Value
f_1	230 N/m ³
f_2	4580 N/m ³
f_3	1470 N/m ³
f_4	290 N/m ³
g_1	18 900 N/m ^{2.5} /s
g_2	670 N/m ³ /s
g_3	1550 N/m ^{3.5} /s

Table B.3: Constants in Riska resistance model, courtesy of Hu and Zhou (2016).

Ship Parameters	V	Ship speed
	L	Ship length
	B	Ship beam
	T	Ship draft
	L_{bow}	Bow length
	L_{par}	Parallel mid-body length
	φ	Stem angle
Ice parameters	h	Ice thickness
	σ_f	Flexural strength ¹
	μ_{si}	Structure-ice friction coeff ² .

Table B.4: Table of parameters required for the Riska resistance model.

¹Value of 500 kPa incorporated in coefficients.

²Value of 0.15 incorporated in coefficients.

Oden Data

L	Ship waterline length	93.2 m
B	Ship beam	29.2 m
T	Ship draft	8 m
DW	Ship dead weight	11 927 005 kg
L_{bow}	Bow length	30 m
L_{par}	Parallel mid-body length	28 m
β	Average buttock angle at waterline	0.43 rad
Ψ	Average bow flare angle at waterline	1.159 rad
φ	Stem angle	0.34 rad
α	Waterline angle	0.379 rad

Table C.1: Vessel parameters for Oden.

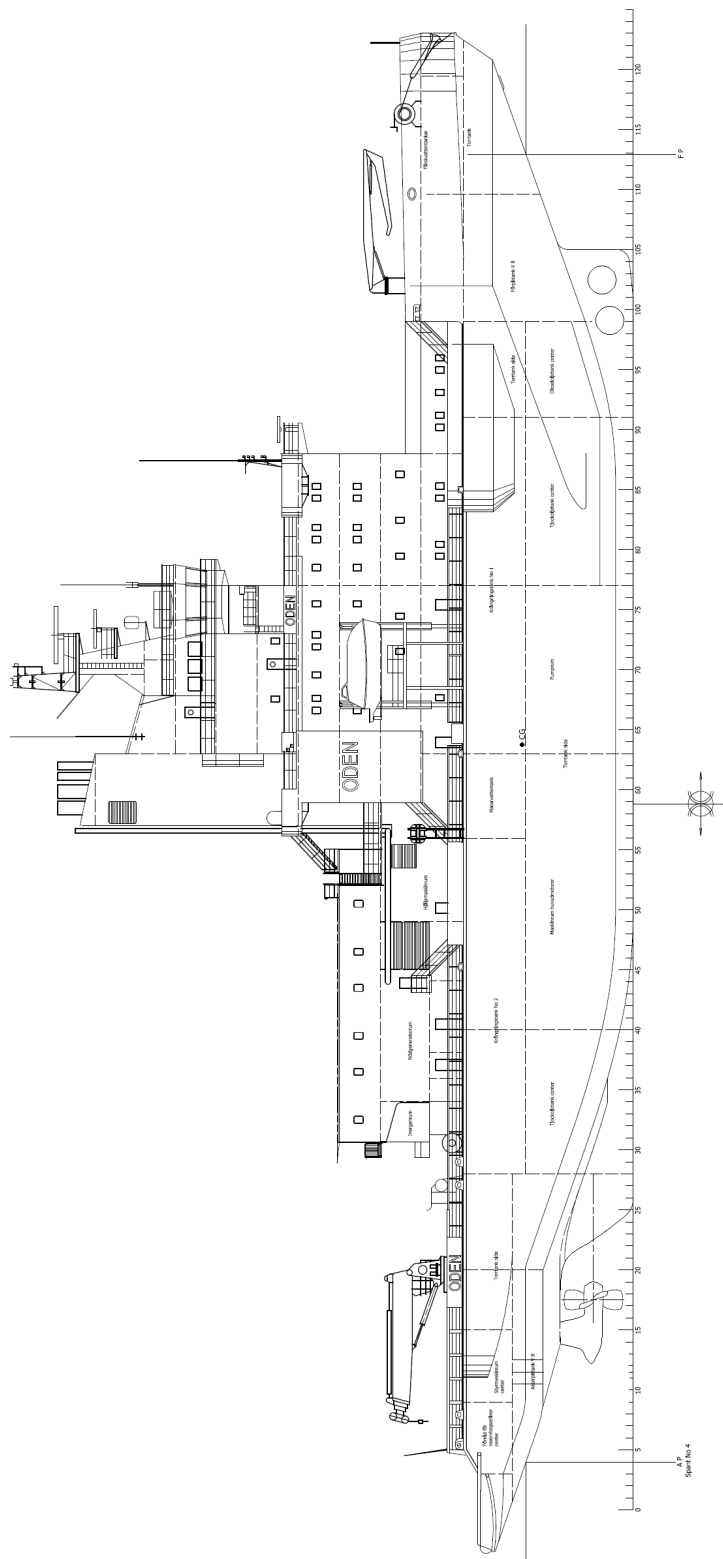


Figure C.1: Blueprint showing the modified thruster position of Oden.

Results

D.1 Observer Verification

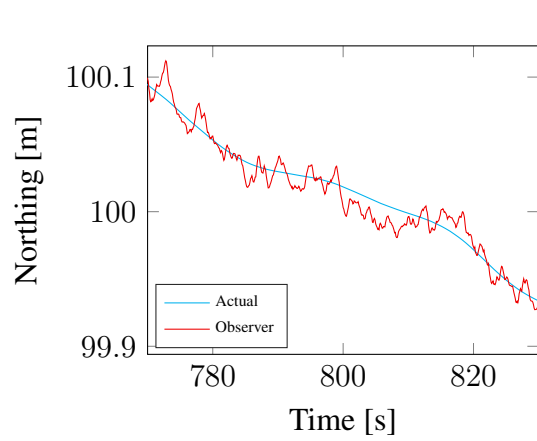


Figure D.1: Scaled version of Figure 7.4.

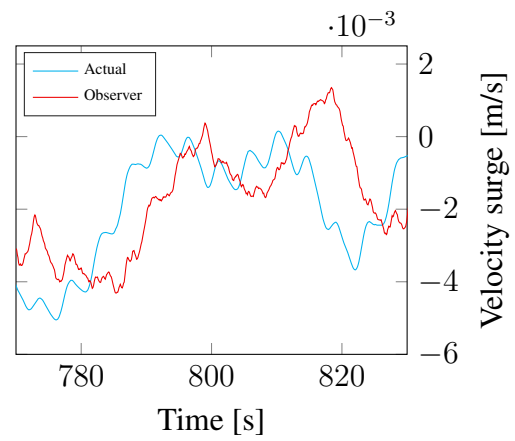


Figure D.2: Scaled version of Figure 7.5.

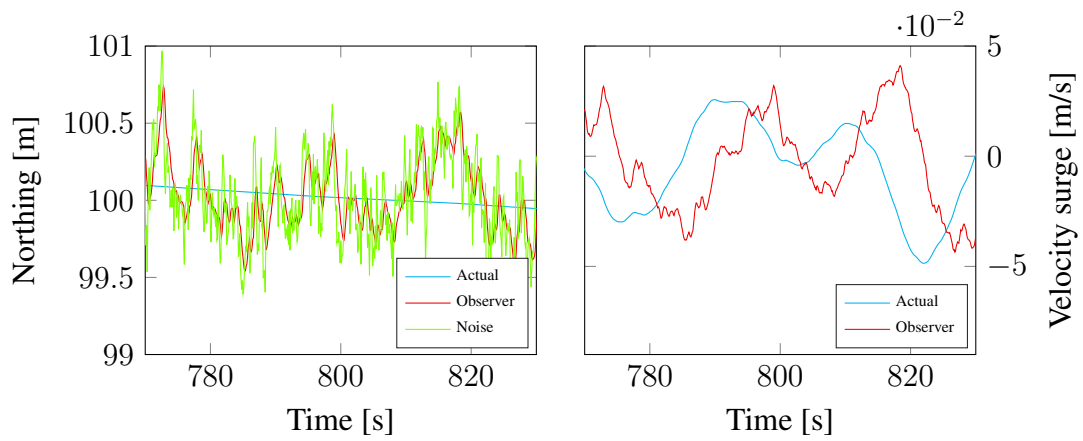


Figure D.3: Actual, estimated and unfiltered northing positions with 20 times normal white noise. **Figure D.4:** Actual and estimated velocities with 20 times normal white noise.

D.2 OATRC 2015 Transit Simulation

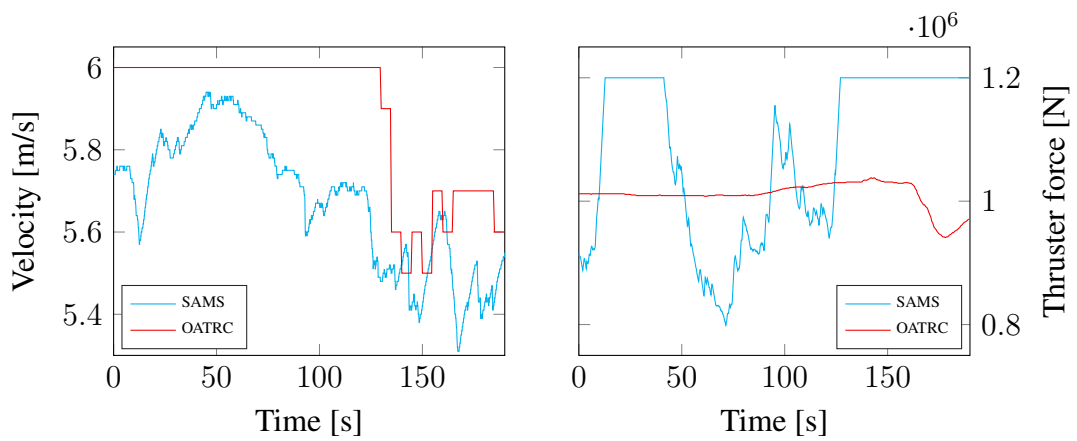


Figure D.5: Actual and simulated surge velocities from OATRC simulation. **Figure D.6:** Predicted and simulated thruster forces in surge from OATRC simulation.

D.3 Transit Acceleration Feedback Simulation

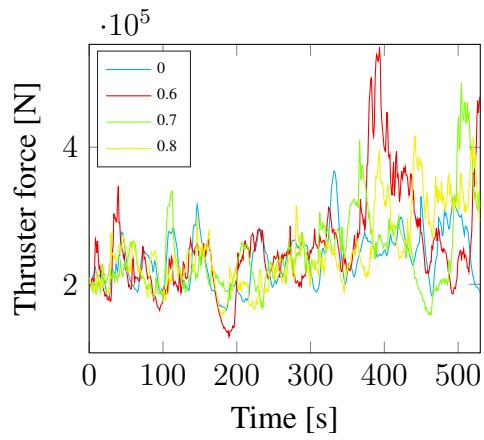


Figure D.7: Filtered thruster forces in surge from acceleration FB simulations.

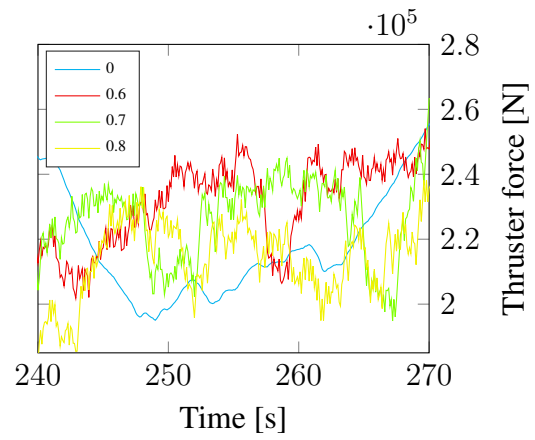


Figure D.8: Unfiltered thruster forces in surge in a smaller time interval from Figure D.7.

D.4 Dynamic Positioning without Ice Management

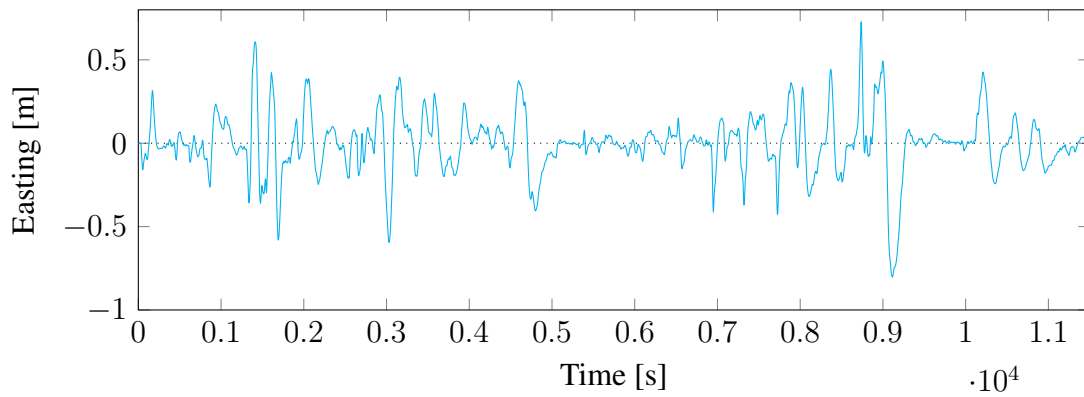


Figure D.9: Easting position of Oden during DP simulation without IM support.

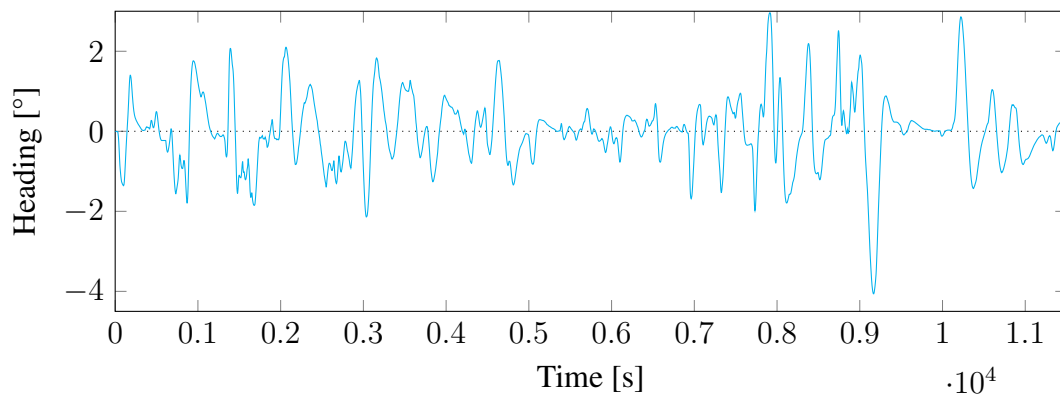


Figure D.10: Heading of Oden during DP simulation without IM support.

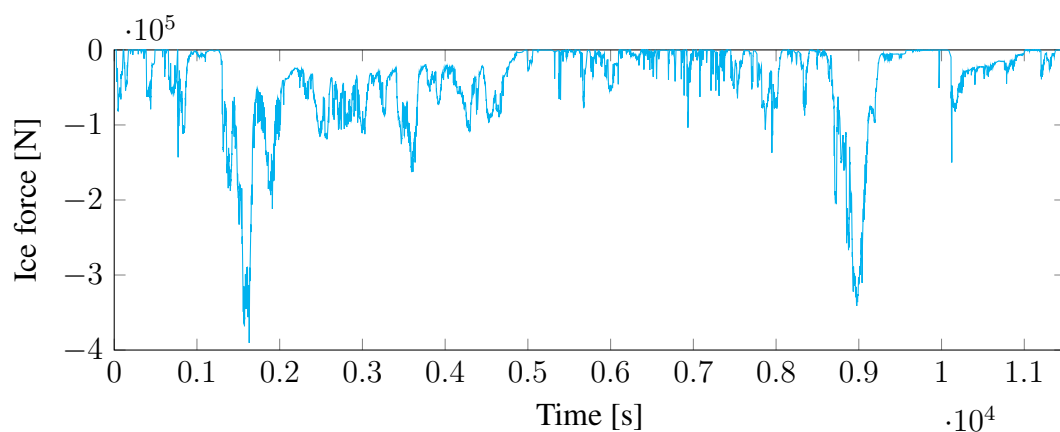


Figure D.11: Filtered ice force in surge of Oden during DP simulation without IM support.

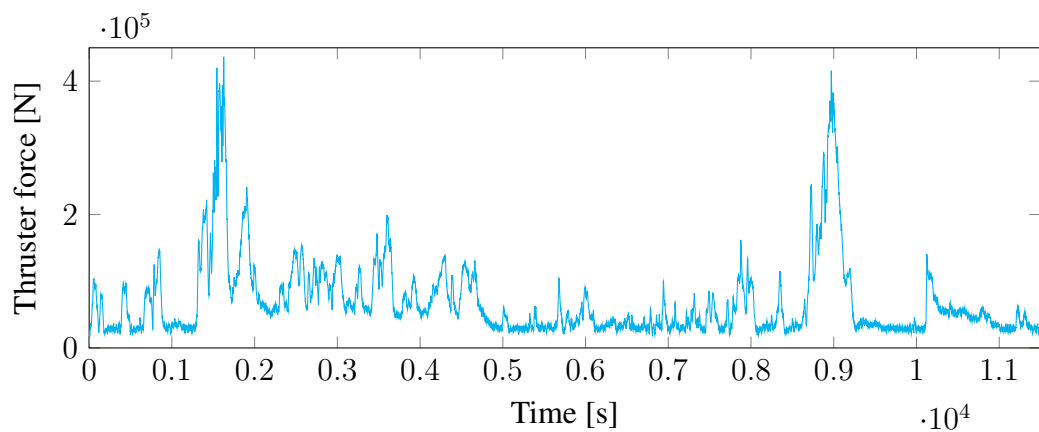


Figure D.12: Filtered thruster force in surge of Oden during DP simulation without IM support.

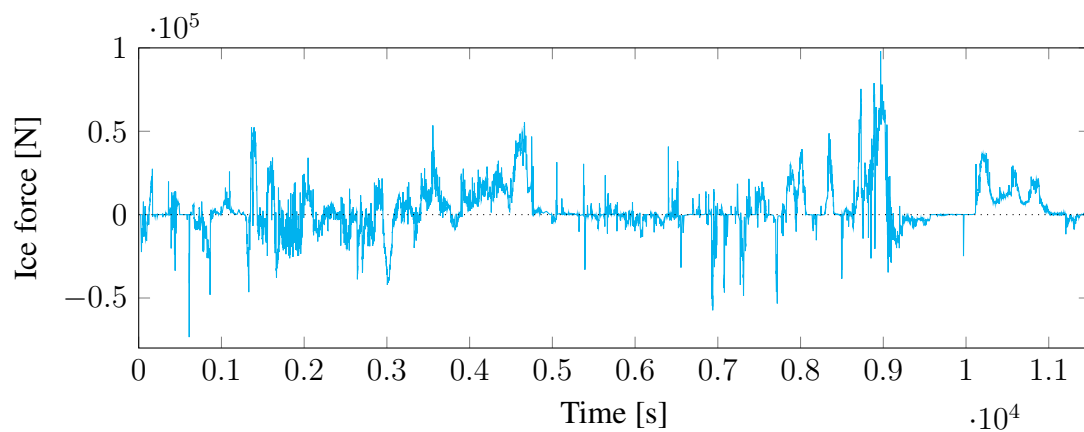


Figure D.13: Filtered ice force in sway of Oden during DP simulation without IM support.

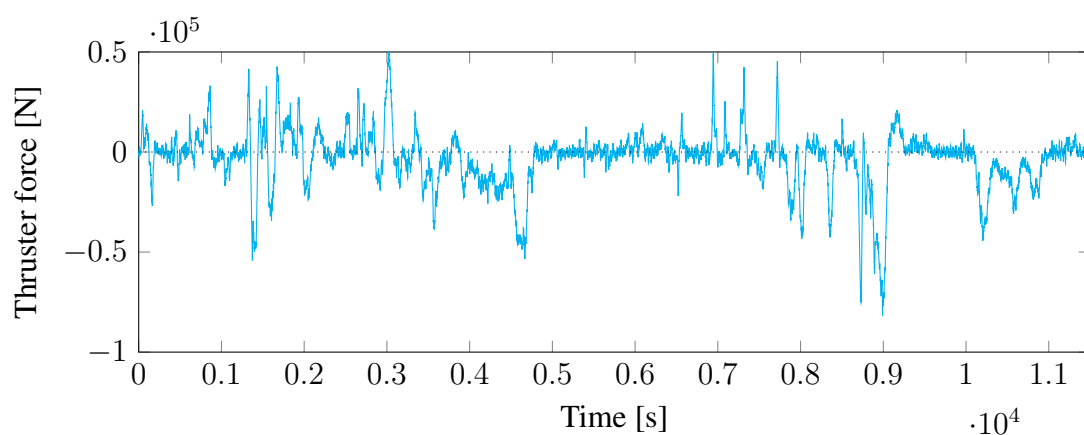


Figure D.14: Filtered thruster force in sway of Oden during DP simulation without IM support.

D.5 Dynamic Positioning with Ice Management

D.5.1 Dynamic Positioning Vessel

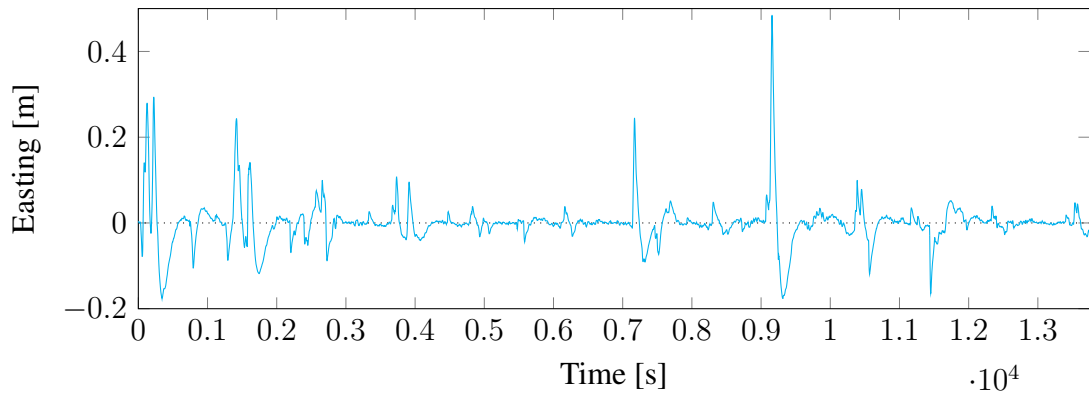


Figure D.15: Easting position of Oden during DP simulation with IM support.

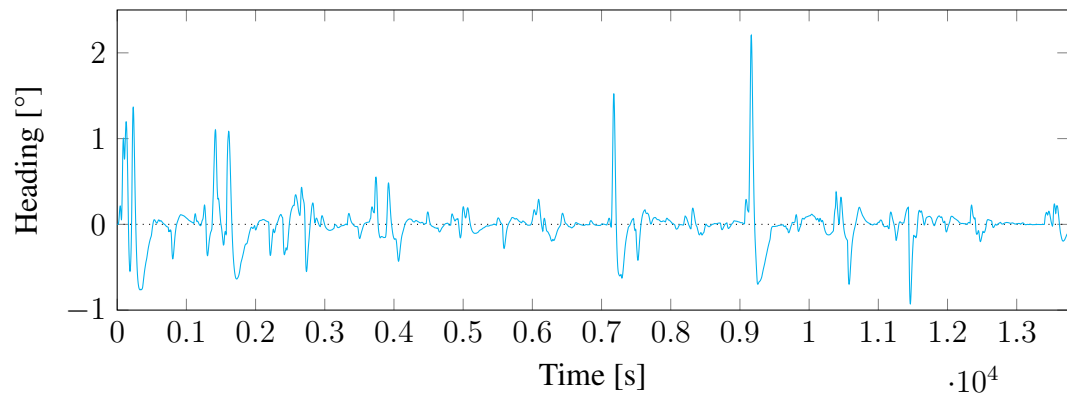


Figure D.16: Heading of Oden during DP simulation with IM support.

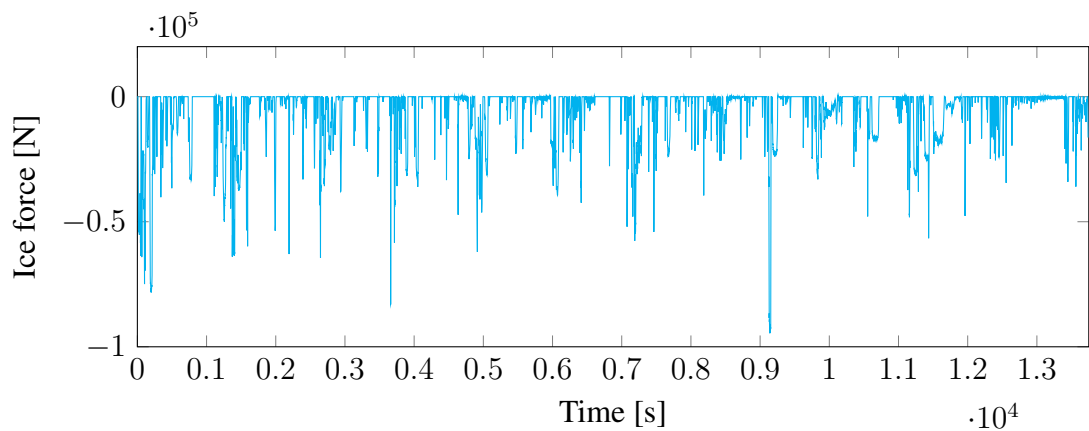


Figure D.17: Filtered ice force in surge of Oden during DP simulation with IM support.

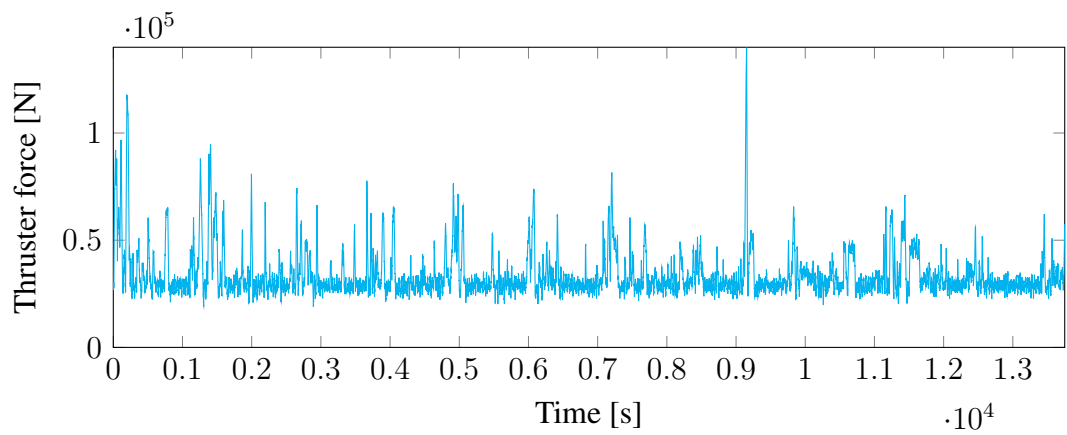


Figure D.18: Filtered thruster force in surge of Oden during DP simulation with IM support.

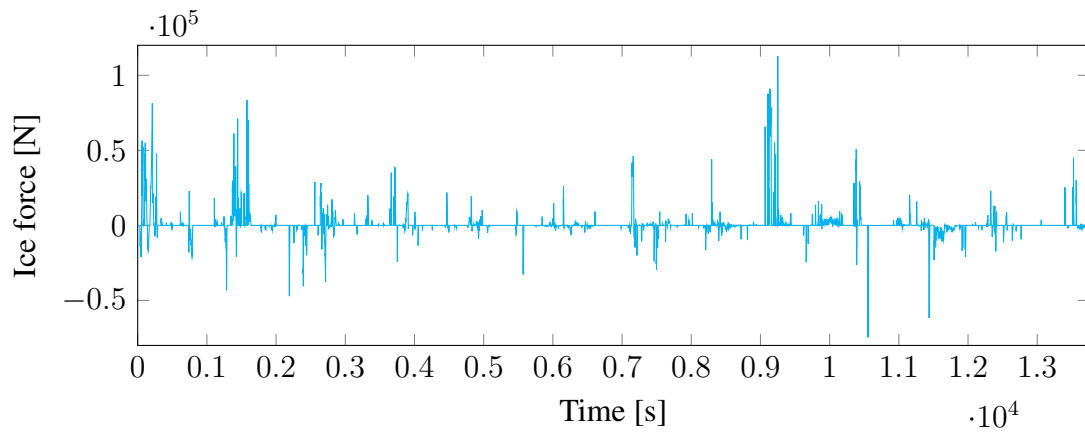


Figure D.19: Filtered ice force in sway of Oden during DP simulation with IM support.

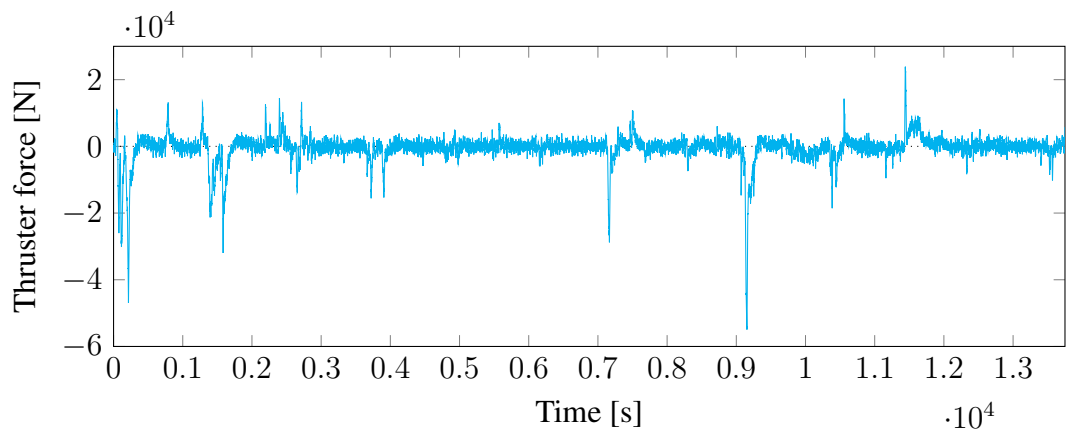


Figure D.20: Filtered thruster force in sway of Oden during DP simulation with IM support.

D.5.2 Ice Management Vessel

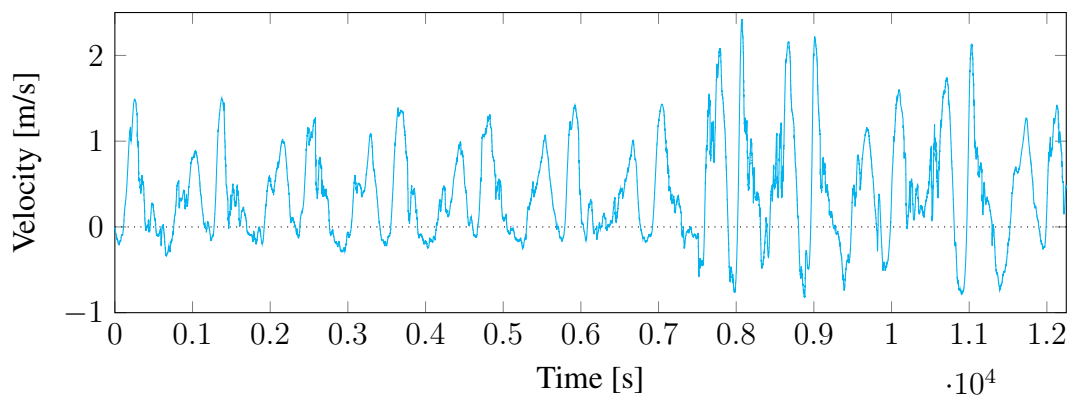


Figure D.21: Sway velocity of IM vessel during DP simulation with IM support.

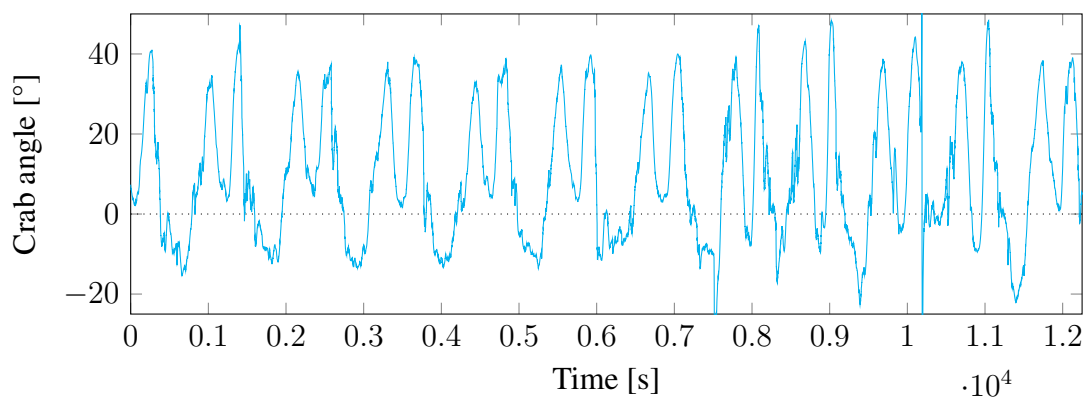


Figure D.22: Crab angle of IM vessel during DP simulation with IM support.

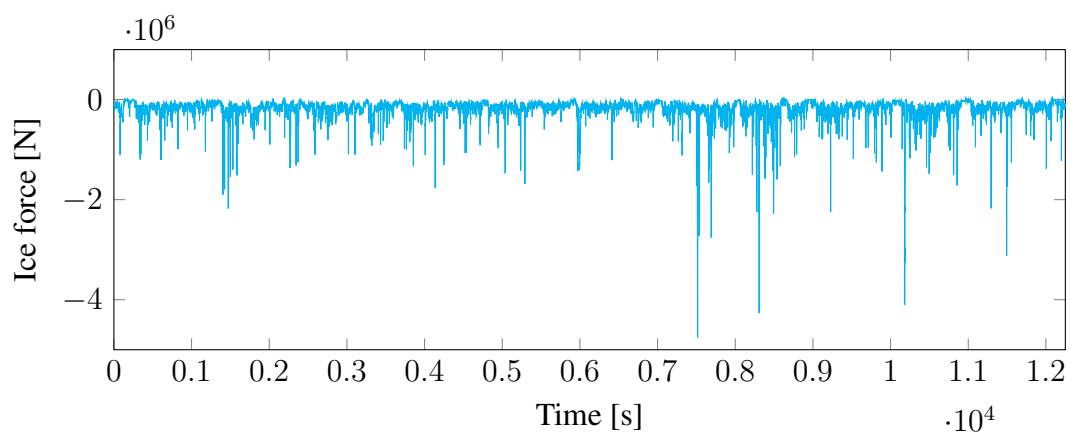


Figure D.23: Filtered ice force in surge of IM vessel during DP simulation with IM support.

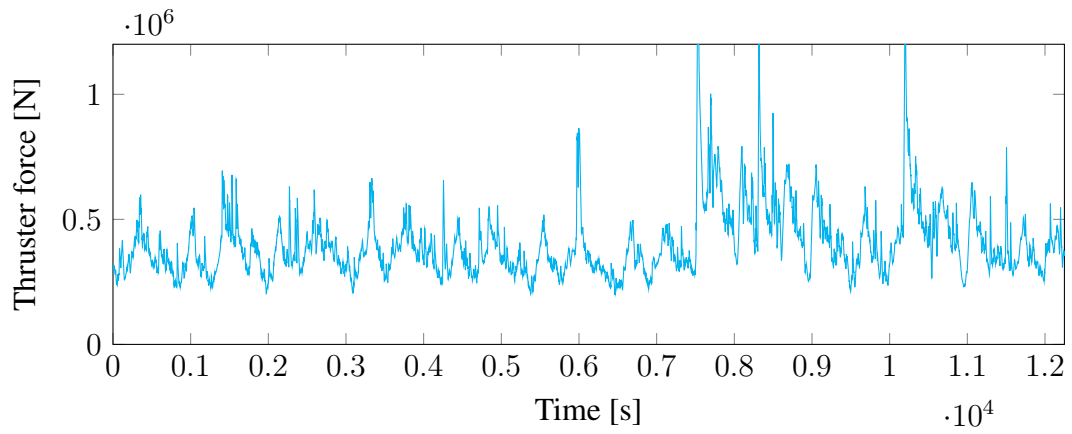


Figure D.24: Filtered thruster force in surge of IM vessel during DP simulation with IM support.

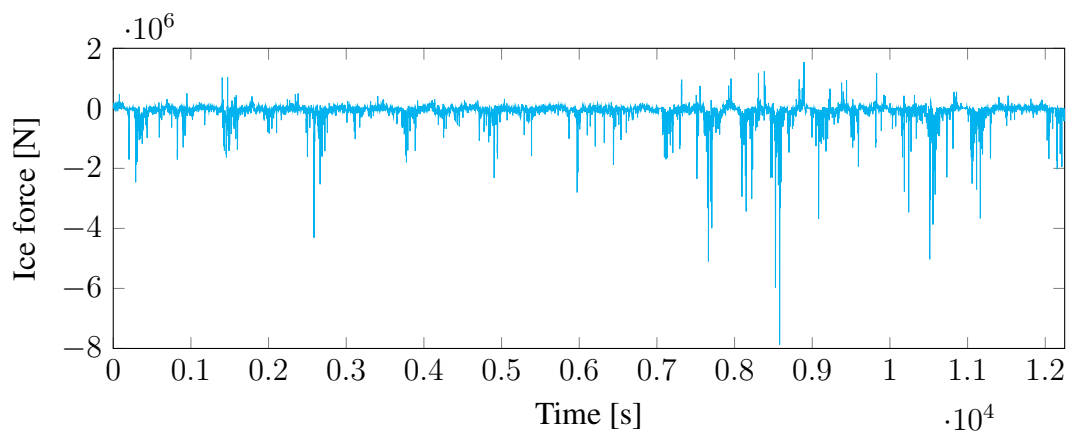


Figure D.25: Filtered ice force in sway of IM vessel during DP simulation with IM support.

THIS PAGE INTENTIONALLY LEFT BLANK

Documentation

E.1 *RHAS.itconfig*

This configuration file is read in after the *SAMS.itconfig* file and is used for the additional properties needed for free moving structures in SAMS. The file consists of several JSON objects where the first two contains parameters for setting up and controlling the simulation environment, then the following JSON objects define the additional structures.

RHASProperties

The first JSON object *RHASProperties* has eight members:

- "enableTowingCarriage" : *Boolean*
- "portNumber" : *Integer*
- "numberOfRHASShips" : *Integer*
- "useJSON" : *Boolean*
- "useCustomHydrodynamics" : *Boolean*
- "initializeStructureAtRest" : *Boolean*
- "iceFloesInitVelocityX" : *Double*
- "iceOutputWaterVelocity" : *Boolean*

"enableTowingCarriage" control whether the towing carriage should be activated or not, setting this to true disables the TCP interface.

"portNumber" specifies the port for the first vessel, any additional vessels will have this port + i , where i is an integer going from one and increasing by one for every additional vessel. The 5th vessel will for example have port number "portNumber" + 4.

”NumberOfRHASShips” specify how many extra vessels should be initialized. Note that one vessel is always initialized from *SAMS.itconfig*. That means that if three vessels are wanted two should be specified here. Likewise if zero is specified only the vessel initialized in *SAMS.itconfig* will be initialized.

”UseJSON” enables or disables the use of RapidJSON as serialization format for encoding and decoding of the TCP data. For use with Simulink this should be set to false. Setting this to false means the data will be sent as a bytestream, for information on how this bytestream looks like please see Section E.2. This should be set to true if the manual web application is used, note that multiple vessels and custom hydrodynamics are not possible in the web application.

”useCustomHydrodynamics” enables or disables the hydrodynamics acting on the vessels. When set to true the simulators own hydrodynamic calculations for the vessels are disabled, and all hydrodynamic forces must be sent in to the simulator as six double precision values (three forces and three moments). Note that the simulator still calculate hydrodynamic forces on ice bodies. When set to false the simulator calculates the hydrodynamic forces acting on each vessel on its own. Note that this function only works when ”useJSON” is set to false.

”initializeStructureAtRest” enables or disables the calculation of the static rest position of the vessels. When put to true the simulator will calculate the exact static rest position and move the structure to this position.

”iceFloesInitVelocityX” give the ice floes a start velocity in surge equal to the specified velocity. This is used when relatively large current velocities are simulated and no build up time for the current velocity is possible.

”iceOutputWaterVelocity” enables or disables the logging of the water velocity vectors. These vectors are used in *IceView* for visualization of the simulation, where they are great for visualization how the thrusters operated during a simulation. Note that not much additional time is needed for logging of these vectors, but the output files quickly becomes large.

IceFloeRemoval

The second JSON object *IceFloeRemoval* controls the removal of ice floes that go outside a designated scope. The main use for this object is when current is enabled for large ice fields, as ice floes that drift outside the scope/tank can be deleted and thus save computational power. The object has three members:

- ”enable” : *Boolean*
- ”xPosition” : *Integer*
- ”timestep” : *Float*

”enable” enables or disables this custom removal of ice floes. When disabled the simulator works as without the ice floe removal and none of the other members have any influence.

”xPosition” controls the position for where to remove ice floes. All ice floes that have a center position lower than this value will be removed. Note that it only looks at the center position of the ice floe, precaution should thus be taken to avoid removing ice floes that could still be of interest.

”timestep” controls how often this removal should happen, the time step is given in seconds. To save computation this can be set to a higher number, if the removal should be run each time step this can be set to the time step of the simulator.

Structures

The following JSON objects is for the structures. Since multiple structures can be initialized (all in separate objects) all needs unique names. This is achieved by adding the structure number after the object name, i.e. structureProperties1, structureProperties2, structureProperties3 Note that there can be more of these structure-objects than ”numberOfRHASShips”, in that case only the n first structures will be initialized. Note however that there must never be fewer structures than specified in ”numberOfRHASShips”. The JSON object itself is the same as that found in *SAMS.itconfig* note however that here the values of ”axis_on_rb_X”, ”axis_on_rb_Y”, ”axis_on_rb_Z” and ”propeller_thrust” have nothing to say for the results. Note also that ”enableThrustApplication” should be set to true, if not no thruster forces can be put on the structure.

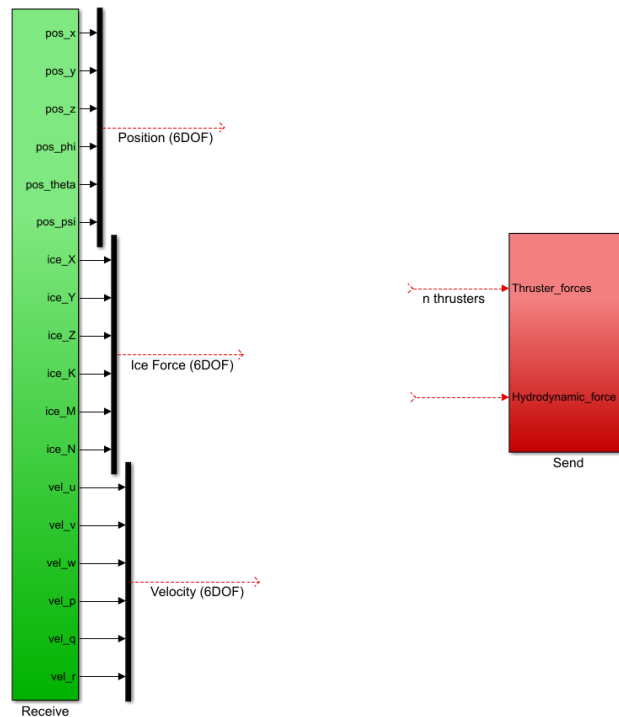


Figure E.1: Overview Simulink template.

E.2 Simulink

Looking at the overview picture in Figure E.1 it shows a template of how the Simulink model is set up. In between these two blocks seen here a control system can be added, where the green block is the receiving end from SAMS and the red block sends the calculated thruster forces and hydrodynamic forces to SAMS. Note that the force given in to the thruster force should be a vector (mux) inside a vector (mux) with three double precision values for the number of thrusters specified for the vessel. The values from this should be given in the BODY-reference system. For all the received values these are given in NED coordinate system. Note if "useCustomHydrodynamics" is enabled, six additional forces needs to be added at the end, this will be sent in the same way as thruster forces and given in a NED-reference system, note that the three first are linear forces and the last three are moments. The system should be made continuous with fixed step size. To change the port number or the address for the computer to connect to one can double click one of the blocks in the subsystem. Then something like Figure E.2 can be seen. Here one can double click TCP/IP Receive/Send and a dialog box like the one in the figure will show up. Here the IP (remote) address and port number can be changed, also the timeout value can be specified. Note 127.0.0.1 is an address that specify that it should connect to the current computer or "localhost" as it is also called. For the receive block the block sample time must be set, and this should be equal to the frequency specified in the *SAMS.itconfig* file. To make sure everything is run in the right order (this is essential when running multiple vessels) one should right

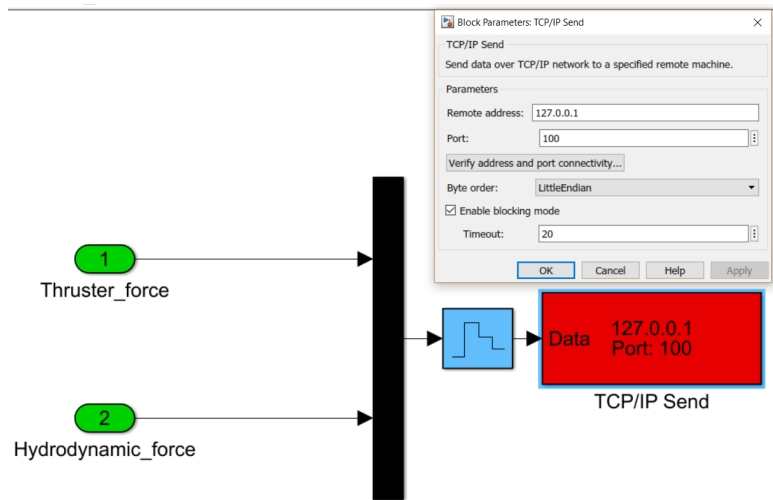


Figure E.2: Inside the "send" block from Figure E.1.

click the systems in the overview and go to properties. Here the priority should be set, where the receive should have lower priority that the send, and vessel one lower than the subsequent additional vessels. Multiple vessels can easily be supported by copy pasting the template and changing the port number and priority like mentioned over.

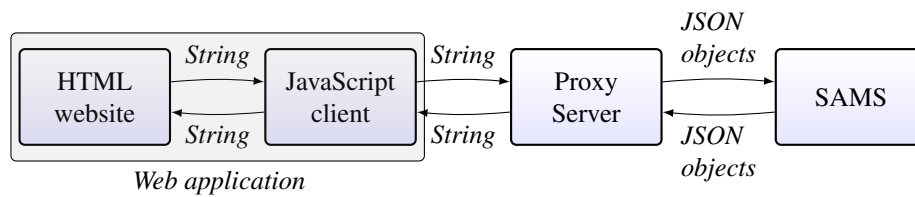


Figure E.3: Web application signal flow.

E.3 Web Application Documentation

About

This web application is for controlling one modified vessel in the SAMS simulator with the different controllers in the web application, where each slider/joystick controls one thruster. The communication between the web application and the simulator is happening over a TCP/IP connection and thus do not need both to be run on the same computer, note that this application is more of a demonstration and proof of concept than for any useful control application of the vessel (however with further development this can for instance be used as a training tool for captains in the future). The application sends JSON objects of forces calculated from the joystick and slider bar in the web application, and is also receiving JSON objects from the SAMS simulator consisting of the position and velocity of the vessel. To do this a proxy server has been created backend with NODE.js, this is where the conversion from the joystick- and slider values to forces and further creates a JSON object for sending to SAMS. JSON objects coming from the simulator with position and velocity is also received in the proxy server before it is sent to the web application, see Figure E.3.

<https://nodejs.org/en/>

<https://www.json.org/>

Prerequisites

- Windows 64-bit.
- SAMS simulator with the TCP interface implemented.
- Node.js for the proxy server.
- Browser supporting HTML and JavaScript.

Setup

There are three steps in the setup of this web application:

Firstly the simulator has to be setup. In the RHAS.itconfig file *enableTowingCarriage* has to be set to false (to disable the towing carriage and use an external control system),

```
C:\Users\Andre\Documents\Prosjektoppgave\git\SAMS_TCP\SAMS\WebApp>node proxy.js
Specify the port which the WebApp should use (<Enter> for default: 99):
Specified port is: 99
Specify the IP-adress for the simulator (<Enter> for default: 127.0.0.1):
The adress is set to: 127.0.0.1
Specify the remote port which the simulator should use (<Enter> for default: 100):
Specified port is: 100
redirecting connections from 127.0.0.1:99 to 127.0.0.1:100
```

Figure E.4: Illustration of setup of proxy server.

portNumber needs to be specified (this is the port number the proxy server connects to), and lastly *useJSON* needs to be true (this is to send and receive JSON objects).

The second step is to start the proxy server so a connection can be setup, this is done in a command prompt window, and after navigating to the correct file path simply start the proxy server by typing:

```
node proxy.js
```

Three variables are needed for the setup, the first is the port for connecting to the web application (the JavaScript client), the default is set to port 99. The second is the IP-address for the simulator (the IP-address of the computer the simulator is started on), the default address is set to *localhost*. The last variable is the port for the simulator, this needs to be the same as the one set as *portNumber* in the *RHAS.itconfig* file, the default port is 100.

NOTE: Starting the simulator first is essential for the connection to happen, if no connection is made then the proxy server shuts down, also the proxy server and web application needs to be run on the same computer, the setup should look like Figure E.4.

The last step is to connect the web application to the proxy server, this is done by specifying the portnumber as the same as the first port in the setup of the proxy server (not the same as in *RHAS.itconfig*). Now a connection to the proxy server can be made and the simulation can be started.

In order to start another simulation after one is started the simulator, proxy server and the web application needs to be restarted (or refreshed).

E.4 Output

Parameter	Name in file	Unit	Note
Time	T	s	
Position	x, y, z	m	
Attitude	ksi, theta, psi	rad	
Velocity	u, v, w	m/s	
Angular velocity	p, q, r	rad/s	
Crushing ice force	IceF_X_crush IceF_Y_crush IceF_Z_crush	N	
Rigid ice force	IceF_X_rigid IceF_Y_rigid IceF_Z_rigid	N	Not physically correct, needs to be filtered.
Drift velocity	Current_X Current_Y Current_Z	m/s	
Wind velocity	Wind_X Wind_Y Wind_Z	m/s	
Hydrodynamic force	HydroF_X HydroF_Y HydroF_Z	N	In CG. Only if custom hydrodynamics is enabled.
Hydrodynamic torque	HydroT_X HydroT_Y HydroT_Z	N m	In CG. Only if custom hydrodynamics is enabled.
Thruster forces	Thr_i_X Thr_i_Y Thr_i_Z	N	BODY frame thruster position. i is the thruster number.

Table E.1: Output format provided by the simulator.

Parameter	Name in file	Unit	Note
Time	T	s	
Position	x, y, x	m	
Mass	mass	kg	
Outside Scope	Outside Scope	<i>boolean</i>	<i>true</i> if the removed floe was removed outside the specified scope, <i>false</i> if the floe was removed as a result of a low mass.

Table E.2: Output format for the removed floes output file.

THIS PAGE INTENTIONALLY LEFT BLANK

Simulink Model

F.1 Color Coding

Color	Block type	Examples
Green	Sources	Input ports, constants, TCPIP receive
Magenta	From	
Red	Sinks	Output ports, terminators, to workspace
Orange	Goto	
Light blue	Memory	Integrator, derivative, unit delay, state-space, transfer functions, zero-order hold, memory
Grey	Logic	Action port, if, for loop
Yellow	Others	Gain, sum, subsystem, math function, scope

Table F.1: Color coding scheme for Simulink blocks.

F.2 Main Model

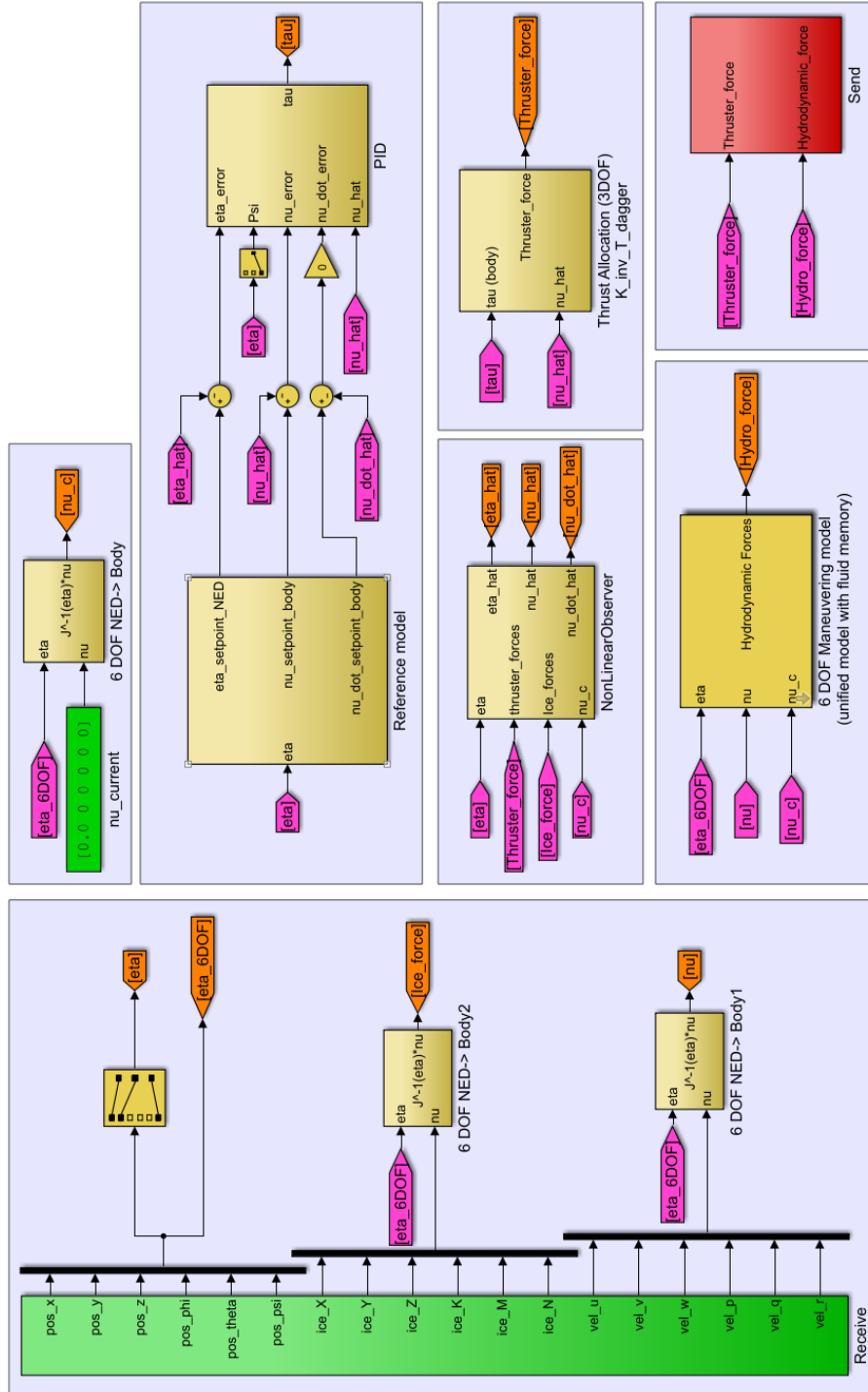


Figure F.1: Overview of main model.

F.3 Reference Models

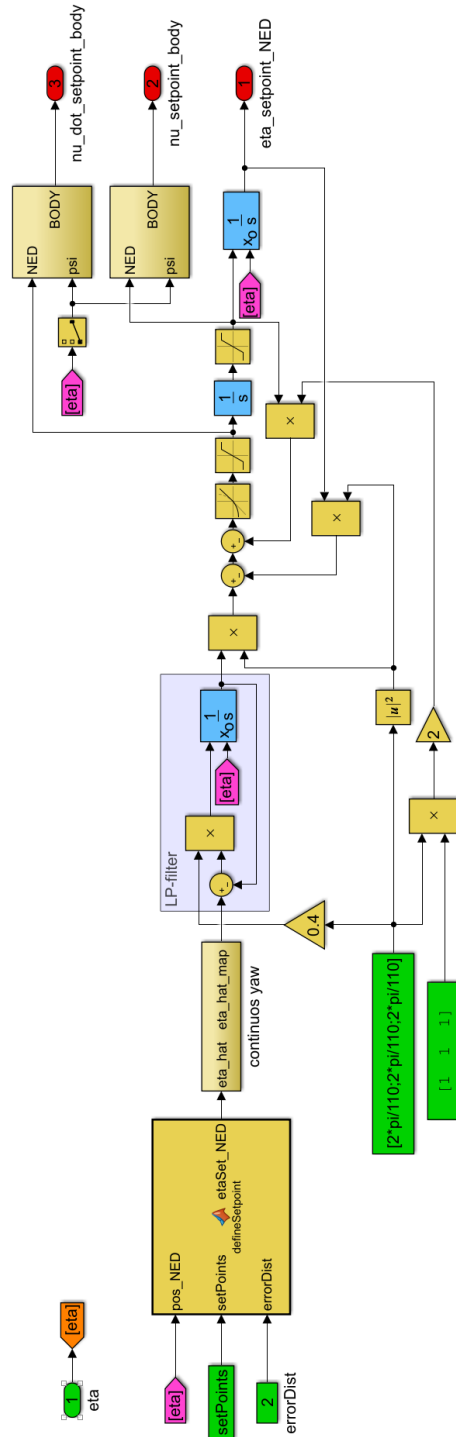


Figure F.2: Overview of reference model.

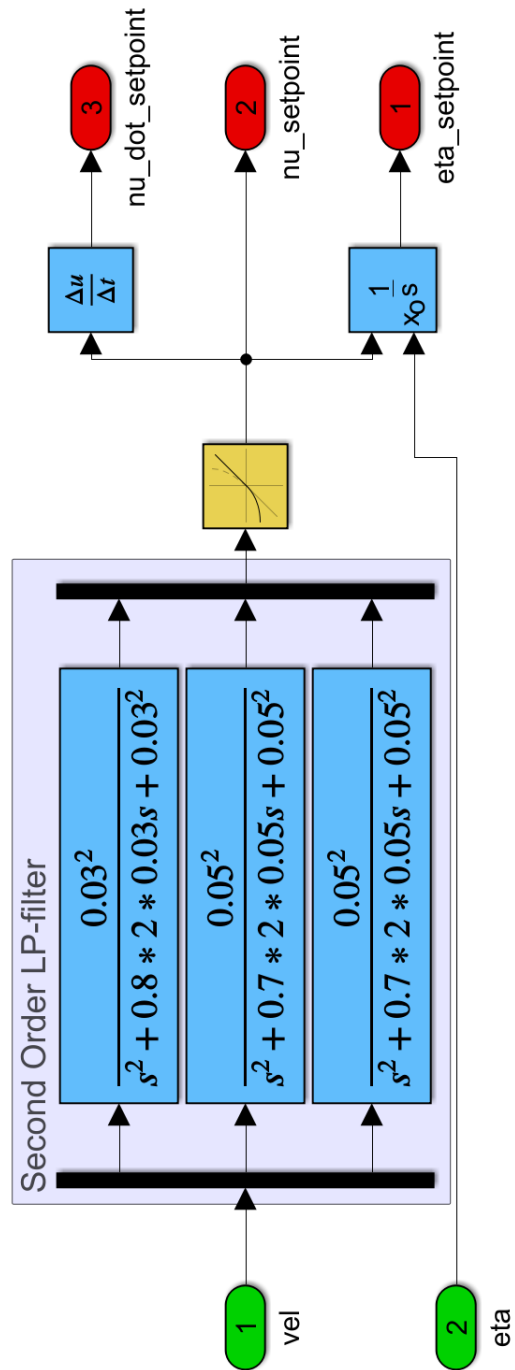


Figure F.3: Overview of velocity reference model.

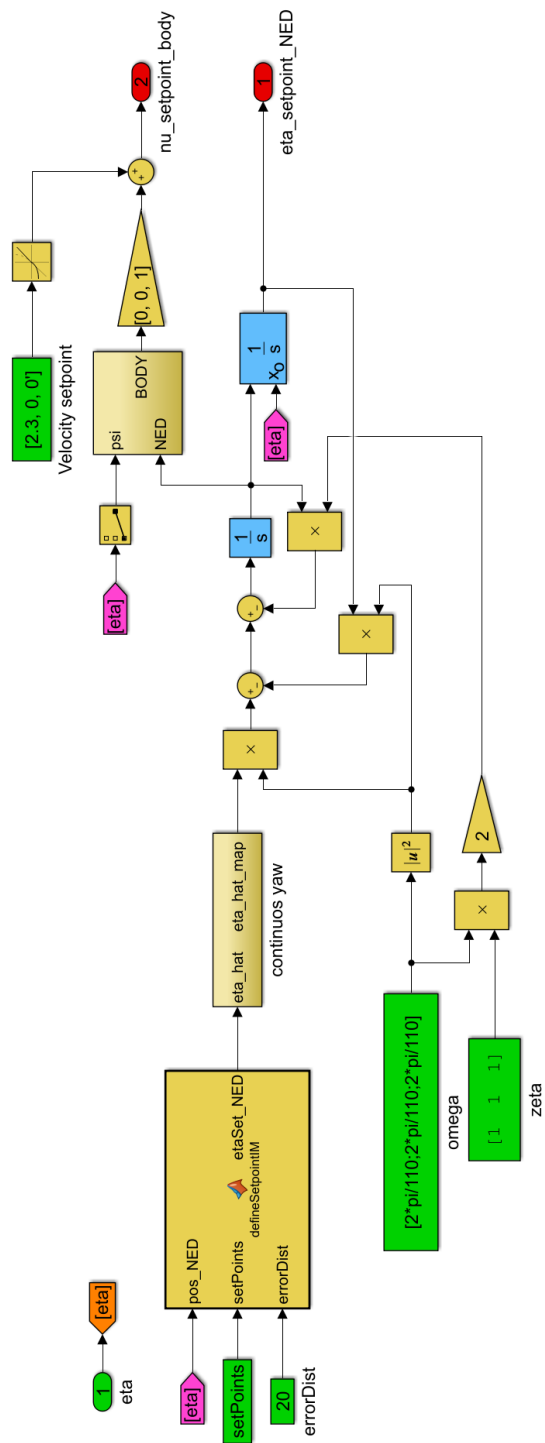


Figure F.4: Overview of IM reference model.

E.4 Observer

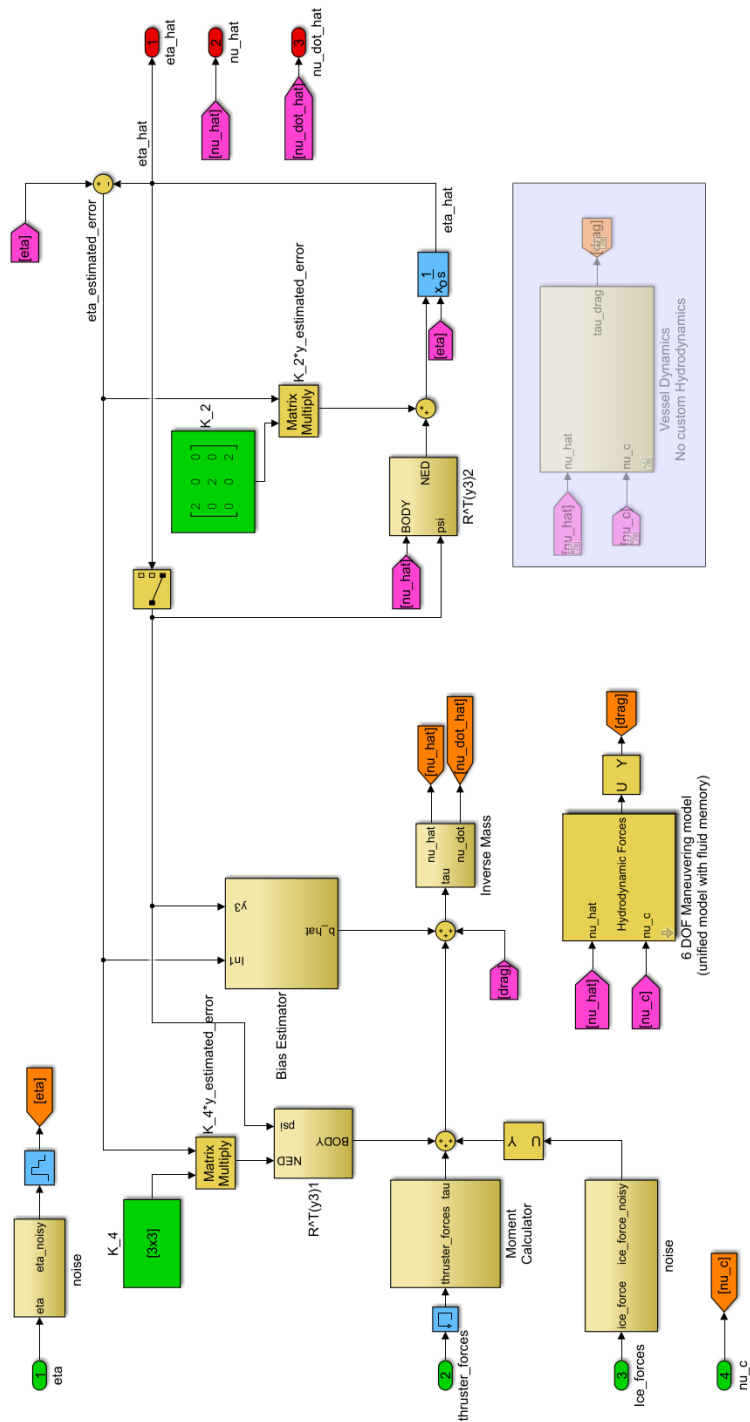


Figure E.5: Overview of observer model.

F.5 Controllers

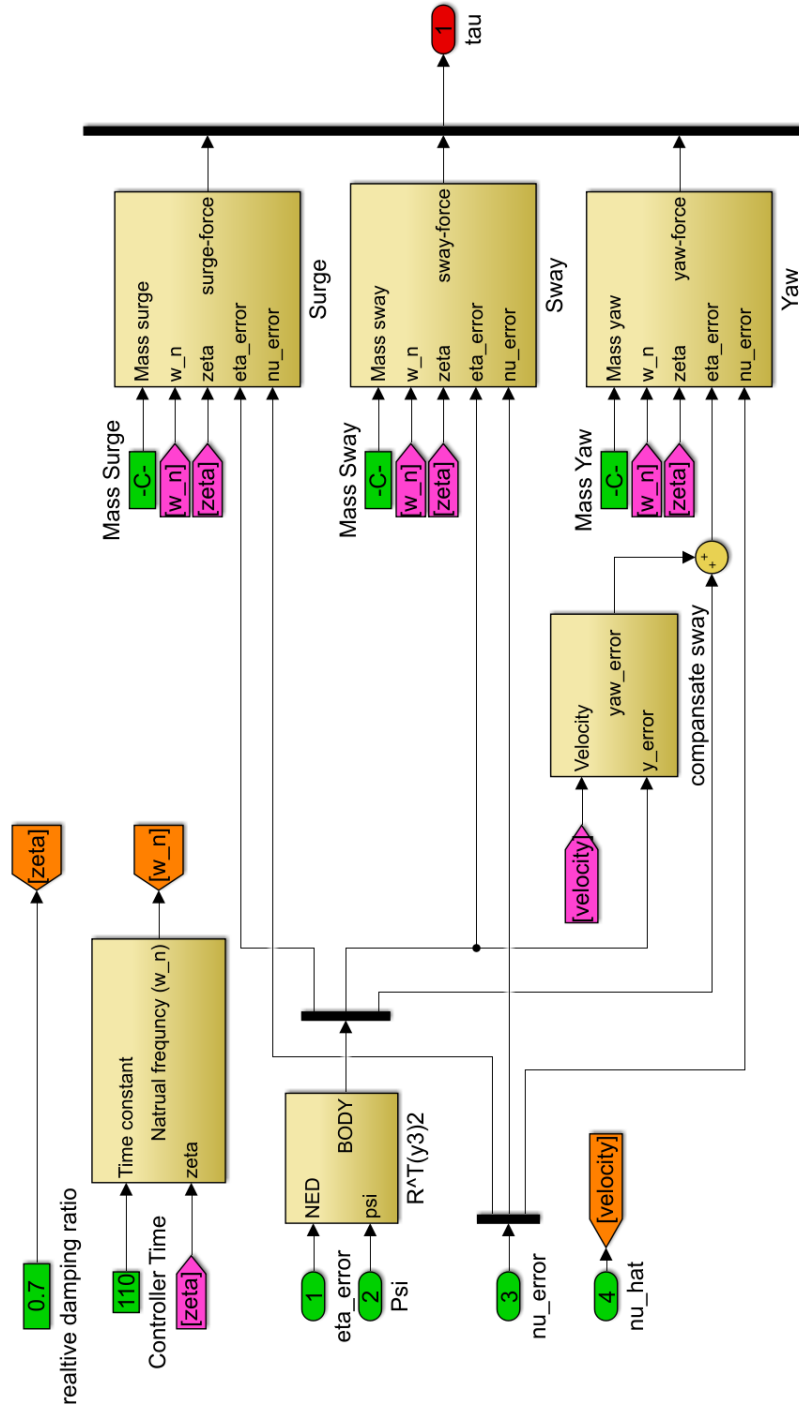


Figure F.6: Overview of DP PID controller.

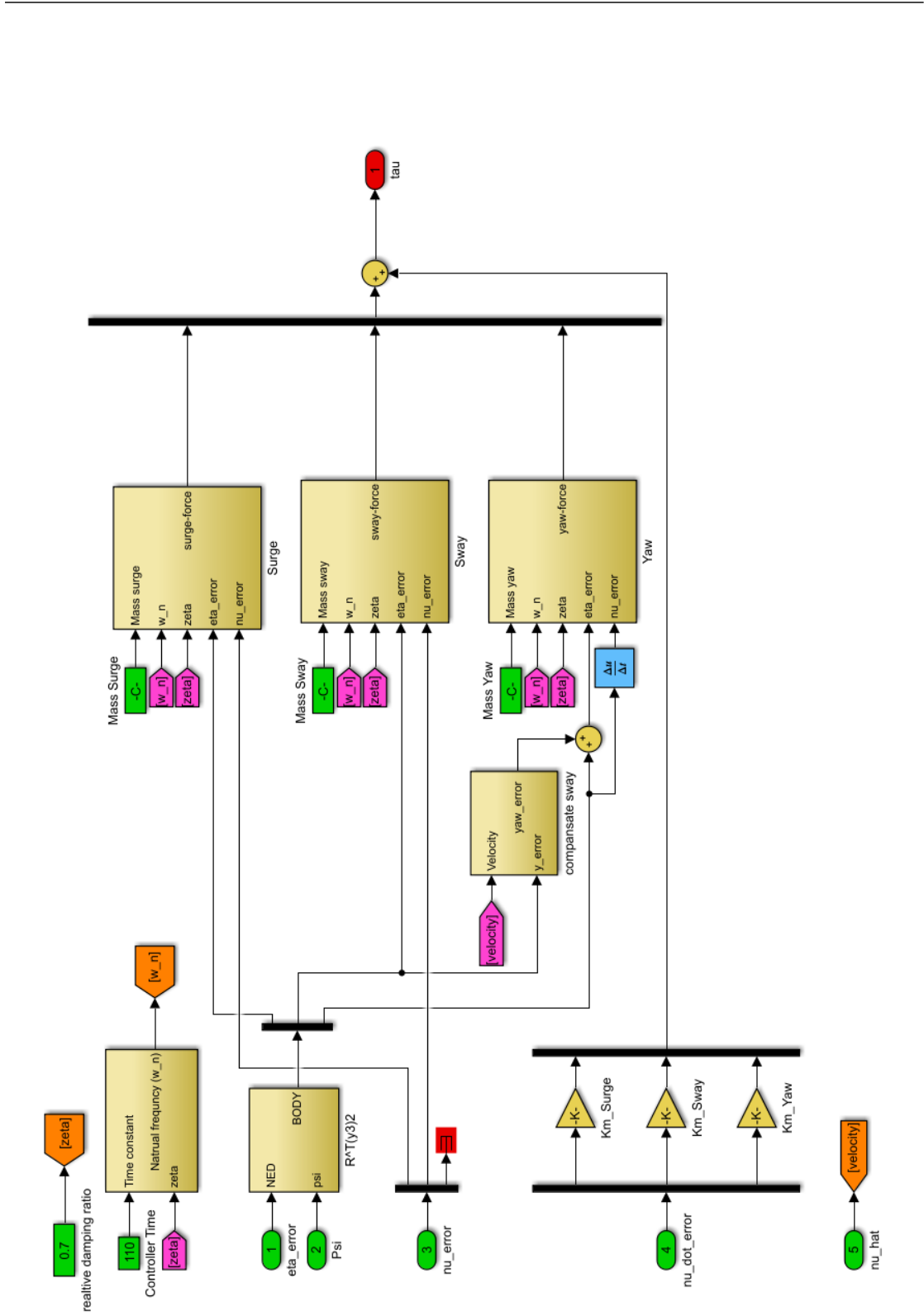


Figure F.7: Overview of acceleration FB controller.

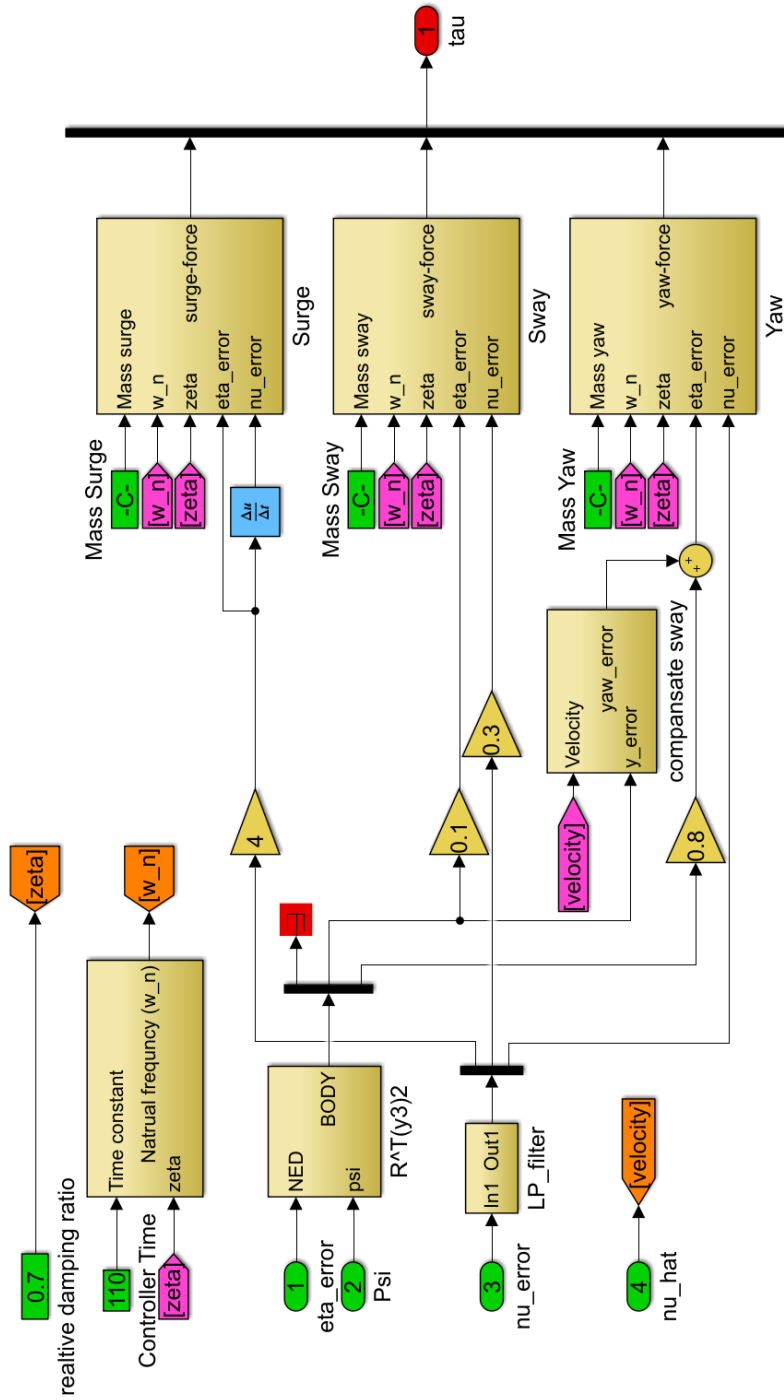


Figure F.8: Overview of IM controller.

E.6 Thrust Allocation

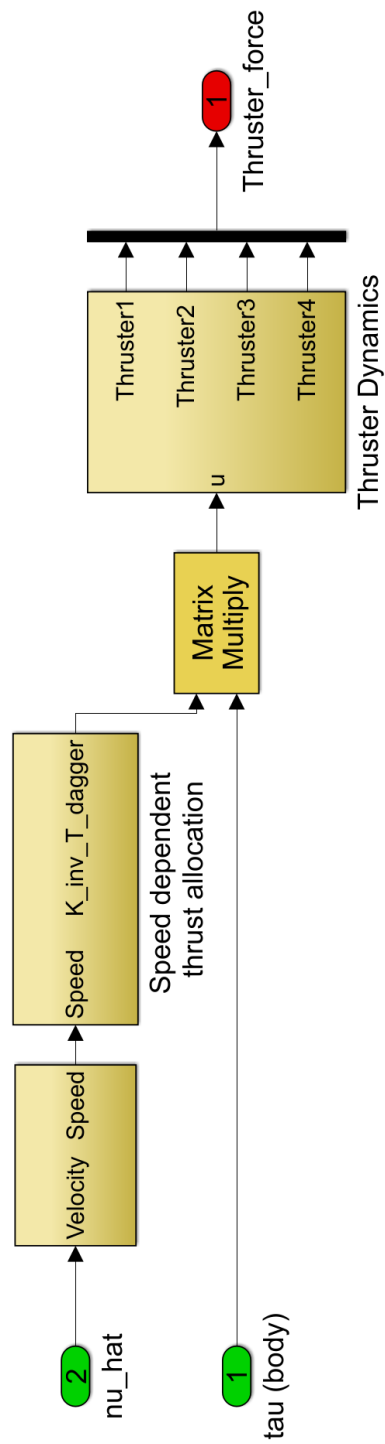


Figure F.9: Overview of thruster allocation model.

E.7 Hydrodynamics

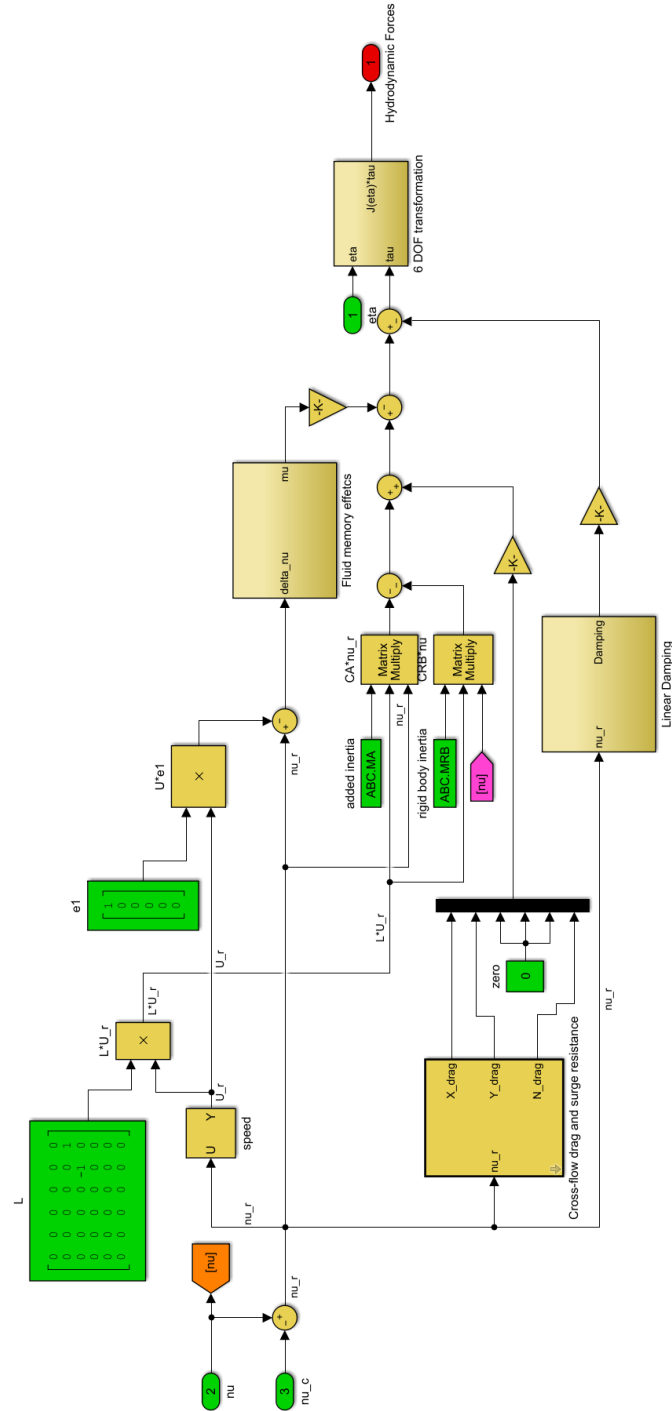


Figure F.10: Overview of the hydrodynamics model from MSS toolbox.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix **G**

Code

G.1 *SAMS.itconfig*

```
1 {
2   "simulationSettings": {
3     "frequency": 10,
4     "enableFracturing": true
5   },
6
7   "iceFieldProperties": {
8     "iceDataFile": "IceField.ice",
9     "waterDepth": 380.0,
10    "keepTankWalls": true,
11    "iceDensity": 900.0,
12    "crushingEnergyAbsorption": 2.0E6,
13    "youngsModulus": 5.0E9,
14    "poissonsRatio": 0.3,
15    "fractureTouchness": 1.5E5,
16    "flexuralStrength": 5.0E5,
17    "tensileStrength": 5.0E5,
18    "friction_Ice_Ice": 0.15,
19    "friction_Ice_Structure": 0.15,
20    "friction_Ice_Walls": 0.15,
21    "minFloemass" : 100000
22  },
23
24  "structureProperties": {
25    "structureObjFile": "Oden_decomposed_05-04-18.obj",
26    "structureMass": 11927005,
27    "structureRadiusOfGyration": [7.45, 30.9, 30.3],
28    "structurePosition": [0, 0, 0],
29    "structureRotationAxis": [0, 0, 1],
30    "structureRotationAngle": 0.0,
31    "numberOfThrusters": 4,
```

```

32     "position_on_rb_X": [-42, -42, 30, 33],
33     "position_on_rb_Y": [-5.0, 5.0, 0.0, 0.0],
34     "position_on_rb_Z": [5.0, 5.0, 6.0, 5.5],
35     "axis_on_rb_X": [-1.0, 1.0, 0.0, 0.0],
36     "axis_on_rb_Y": [0.0, 1.0, 1.0, 1.0],
37     "axis_on_rb_Z": [0.0, 0.0, 0.0, 0.0],
38     "propeller_thrust": [0E4, 0E4, 0E4, 0E4],
39     "propeller_diam": [4.0, 4.0, 2.0, 2.0],
40     "enableThrustApplication": true
41 },
42
43 "towingCarriageProperties": {
44     "flexibleCarriage": true,
45     "towingCarriageSpeed": 10
46 },
47
48 "flexibleCarriageProperties": {
49     "springLinearStiffness": [1E6, 1E6, 1E6],
50     "springRotationalStiffness": [80E6, 80E6, 80E6],
51     "linearDampingCoefficientOnStructure": [0.1, 0.1, 0.1],
52     "angularDampingCoefficientOnStructure": [0.2, 0.2, 0.2]
53 },
54
55 "ice3DResultExportSettings": {
56     "enable": true,
57     "outputTimeStepStride": 50
58 },
59
60 "environmentalProperties": {
61     "airDensity": 1.04,
62     "waterDensity": 1005.0,
63     "maxBodyMeshSize": 10.0,
64     "windVelocityVector": [0.0, 0.0, 0.0],
65     "currentVelocityVector": [0.0, 0.0, 0.0],
66     "waterSkinFrictionCoefficient": 0.005,
67     "waterFormDragCoefficient": 0.167,
68     "airSkinFrictionCoefficient": 0.0,
69     "fluid3DMeshSize": [20, 20, 1],
70     "numberOfFluidMeshLayers": 1
71 }
72 }

```

G.2 *RHAS.itconfig*

```
1 {
2   "RHASProperties": {
3     "enableTowingCarriage" : false,
4     "portNumber" : 100,
5     "numberOfRHASShips" : 0,
6     "useJSON" : false,
7     "useCustomHydrodynamics" : true,
8     "initializeStructureAtRest" : true,
9     "iceFloesInitVelocityX" : 0,
10    "iceOutputWaterVelocity" : false
11  },
12
13  "IceFloeRemoval": {
14    "enable" : false,
15    "xPosition" : 0,
16    "timestep" : 10
17  },
18
19  "structureProperties1": {
20    "structureObjFile": "Oden_decomposed_05-04-18.obj",
21    "structureMass": 11927005,
22    "structureRadiusOfGyration": [7.45, 30.9, 30.3],
23    "structurePosition": [-50.0, 0.0, 0],
24    "structureRotationAxis": [0, 0, 1],
25    "structureRotationAngle": 0.0,
26    "numberOfThrusters": 4,
27    "position_on_rb_X": [-42, -42, 30, 33],
28    "position_on_rb_Y": [-5.0, 5.0, 0.0, 0.0],
29    "position_on_rb_Z": [5.0, 5.0, 6.0, 5.5],
30    "axis_on_rb_X": [-1.0, 1.0, 0.0, 0.0],
31    "axis_on_rb_Y": [0.0, 1.0, 1.0, 1.0],
32    "axis_on_rb_Z": [0.0, 0.0, 0.0, 0.0],
33    "propeller_thrust": [0E4, 0E4, 0E4, 0E4],
34    "propeller_diam": [4.0, 4.0, 2.0, 2.0],
35    "enableThrustApplication": true
36  },
37
38  "structureProperties2": {
39    "structureObjFile": "Oden_decomposed_05-04-18.obj",
40    "structureMass": 11927005,
41    "structureRadiusOfGyration": [7.45, 30.9, 30.3],
42    "structurePosition": [50.0, -50.0, 0],
43    "structureRotationAxis": [0, 0, 1],
44    "structureRotationAngle": 0.0,
45    "numberOfThrusters": 4,
46    "position_on_rb_X": [-42, -42, 30, 33],
47    "position_on_rb_Y": [-5.0, 5.0, 0.0, 0.0],
48    "position_on_rb_Z": [5.0, 5.0, 6.0, 5.5],
49    "axis_on_rb_X": [-1.0, 1.0, 0.0, 0.0],
```

```
50     "axis_on_rb_Y": [0.0, 1.0, 1.0, 1.0],
51     "axis_on_rb_Z": [0.0, 0.0, 0.0, 0.0],
52     "propeller_thrust": [0E4, 0E4, 0E4, 0E4],
53     "propeller_diam": [4.0, 4.0, 2.0, 2.0],
54     "enableThrustApplication": true
55 }
56 }
```

G.3 Ice Stitcher

```
1  file = 'IM.ice';
2  filesToBeStitched = [
3      'IM_Stitch1.ice',
4      'IM_Stitch2.ice'];
5
6  //Specify file name for the output
7  outputFileName = 'IM_Stitched.ice';
8
9  var fs = require('fs');
10
11
12  var content = fs.readFileSync(file);
13  var json = JSON.parse(content);
14
15
16  for (var j = 0; j < filesToBeStitched.length; j++)
17  {
18      var content_stitch = fs.readFileSync(filesToBeStitched[j]);
19      var json_stitch = JSON.parse(content_stitch)
20
21
22      var lastBodyNumber =
23      → json.timeSteps[0].rigidBodies[json.timeSteps[
24      → 0].rigidBodies.length - 1].bodyId + 1;
25      var numBodiesStitch =
26      → json_stitch.timeSteps[0].rigidBodies.length;
27
28      for (var i = 0; i < json_stitch.timeSteps[0].rigidBodies.length;
29      → i++)
30      {
31          json_stitch.timeSteps[0].rigidBodies[i].bodyId +=
32          → lastBodyNumber;
33      }
34      var rigidArray = json.timeSteps[0].rigidBodies.concat(
35      → json_stitch.timeSteps[0].rigidBodies);
36      json.timeSteps[0].rigidBodies = rigidArray;
37  }
```

```
32     var valueVec = json.timeSteps[0].globalVariables[
    ↪ 0].valuesVec3.concat(json_stitch.timeSteps[
    ↪ 0].globalVariables[0].valuesVec3);
33     json.timeSteps[0].globalVariables[0].valuesVec3 = valueVec;
34
35     var valueAxis = json.timeSteps[0].globalVariables[
    ↪ 1].valuesAxisAngle.concat(json_stitch.timeSteps[
    ↪ 0].globalVariables[1].valuesAxisAngle);
36     json.timeSteps[0].globalVariables[1].valuesAxisAngle =
    ↪ valueAxis;
37
38     var activeBodies = json.timeSteps[0].activeBodyIds.concat(
    ↪ Array.from(new Array(numBodiesStitch), (x,k) => k +
    ↪ lastBodyNumber))
39     json.timeSteps[0].activeBodyIds = activeBodies;
40 }
41
42
43 var obj = JSON.stringify(json, null, 2)
44
45 fs.writeFileSync(outputFileName, obj, 'UTF8');
46
47 console.log('Finnished stitching ice fields, remember to add
    ↪ Domain!\n')
```

THIS PAGE INTENTIONALLY LEFT BLANK

Contents of Digital Appendix

The following files are attached in the digital appendix:

- Digital version of the thesis.
- Digital version of the poster.
- Promotional video showcasing the TCP interface and other improvements.
- Complete Simulink library containing every subsystem developed in this thesis.
- Complete Simulink model containing every vessel model used for the different simulations.
- MATLAB scripts used for initializing the Simulink models.
- MATLAB script used for calculating the level ice and managed ice resistances.
- TCP interface source code.
- Web application source code.
- Ice stitcher source code.