NTNU
Norwegian University of
Science and Technology

Mihhail Afanasjev

2022

Master's thesis

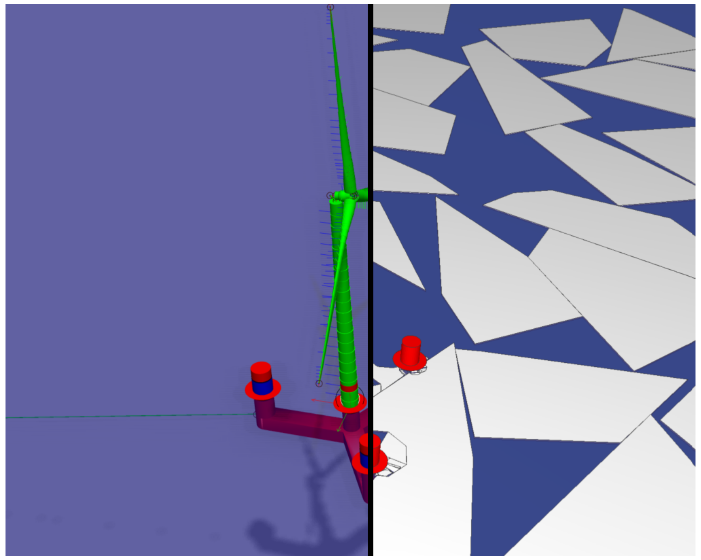
Master's thesis

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

Mihhail Afanasjev

Broken Ice Loads on Floating Wind Turbine

June 2022



NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Broken Ice Loads on Floating Wind Turbine

Mihhail Afanasjev

Nordic Master in Maritime Engineering	
Submission date:	June 2022
Supervisor:	prof. Zhen Gao, NTNU
Co-supervisor:	prof. Wenjun Lu, NTNU prof. Yanlin Shao, DTU

Norwegian University of Science and Technology
Department of Marine Technology

Abstract

In the interest of designing floating wind turbines for cold regions with floating sea ice, the available engineering toolset is expanded in this thesis. SIMA, a marine operations and mooring analysis software, is integrated with SAMS, a simulator for arctic marine structures. With this integration, a flexible, moored to the seabed, floating wind turbine with a blade pitch controller is simulated in a marine environment with wind, waves, current, and a field of broken ice. SIMA solves the main equation of motion. The coupling code applies a coupling force to the structure in SAMS, ensuring that the structures in the two programs have equal position and orientation at each timestep. SAMS finds the ice forces due to contact between the floating platform of the wind turbine and individual ice pieces in the ice field. The ice forces are applied to the SIMA structure.

The coupled model was found to exhibit good robustness of coupling, but high dependence on simulation frequency. A preliminary ice cone design was found to be unsuitable for the scenario of the simulation. The computational cost of the coupled simulation grew rapidly as the simulation frequency was increased.

Abstrakt

I interessen for å designe flytende vindturbiner for kalde områder med flytende havis, utvides det tilgjengelige tekniske verktøysettet i denne oppgaven. SIMA, en programvare for marine operasjoner og fortøyningsanalyse, er integrert med SAMS, en simulator for arktiske marine strukturer. Med denne integrasjonen simuleres en fleksibel, fortøyd til havbunnen, flytende vindturbin med en bladstigningskontroller i et marint miljø med vind, bølger, strøm og et felt med knust is. SIMA løser hovedligningen for bevegelse. Koblingskoden påfører en koblingskraft på strukturen i SAMS, og sikrer at strukturene i de to programmene har lik posisjon og orientering på hvert tidstrinn. SAMS finner iskreftene på grunn av kontakt mellom den flytende plattformen til vindturbinen og individuelle isbiter i isfeltet. Iskreftene påføres SIMA-strukturen.

Den koblede modellen ble funnet å vise god robusthet av kobling, men høy avhengighet av simuleringsfrekvens. En foreløpig iskondesign ble funnet å være uegnet for scenariet for simuleringen. Beregningskostnaden for den koblede simuleringen vokste raskt ettersom simuleringsfrekvensen ble økt.

Acknowledgments

My studies at DTU and NTNU were entirely financed by the Kristjan Jaak Scholarship for Degree Studies Abroad, administered by the Education and Youth Board of the Republic of Estonia. I am deeply grateful for their support and this opportunity. Without it, I would be very unlikely to ever seek any kind of further higher education. My sincerest gratitude goes out to Helen for convincing and helping me to pursue the scholarship.

Professor Yanlin Shao has been exceedingly helpful throughout the two years that I've studied at DTU and NTNU. Even as I moved away to Trondheim, prof. Shao never left me without help when I asked for it - be it coursework, bureaucratic mazes of international joint degrees, or thesis supervision. Yanlin, thank you for these two years.

I found a good friend in Christos. It's funny that we dislike and like exactly opposite things! Without your guidance through these two years, I would have struggled unimaginably more. I hope I was even a fraction as helpful to you as you were to me.

Never would I have imagined that in addition to Maritime departments in two universities, I would find a topic of interest in the Civil Engineering department. Profs. Knut Høyland, Ekaterina Kim, and Raed Lubbad were very welcoming towards me as I ventured into the field of arctic engineering that had been completely foreign to me before. I am especially thankful to professor Wenjun Lu, whose diligent supervision kept me on track and whose good mood never failed to motivate me.

I would like to express gratitude to professors Erin Bachynski-Polić and Zhen Gao, whose abundant patience and meticulous attention to detail helped me crash my way through the wonderful topic of floating wind turbines. Thank you for supervising and helping me!

None of this would ever in a million years have been possible if I didn't know that the people dearest to me believe in me and have my back. Jelena, Natali, Miriam, Konstantin - thank you and I love you.

Table of Contents

List of Figures vi

List of Tables viii

1 Introduction 1

1.1 Floating offshore wind turbines 1

2 Literature review and theory 3

2.1 Semi-submersible floating wind platforms 3

2.2 Ice forces on structures 4

2.2.1 Fracture of ice floes 5

2.2.2 Broken ice 7

2.3 Time-domain methods for calculating broken ice loads on moored structures 8

2.4 Other forces acting on a semi-submersible wind turbine 9

2.4.1 Wind thrust 9

2.4.2 Wind and current drag force 10

2.4.3 Wave excitation forces 10

2.4.4 Buoyancy forces 11

2.5 Newmark-beta time-stepping scheme 11

2.6 Representations of orientation in 3D space 12

2.6.1 Matrix transformation 12

2.6.2 Joint rotations 13

2.6.3 Axis-angle representation 14

2.6.4 Time derivatives 14

3 Method 16

3.1 SIMA 16

3.2 SAMS 16

3.3 Model description 17

3.3.1 Mass and hydrostatic stiffness of the rigid floater 18

3.3.2 Ice-breaking cones 19

3.3.3 Coordinate frames . . . 19
3.4 Equation of motion and coupling force . . . 20
3.5 Coupling code . . . 24
3.6 Coupling algorithm . . . 24
3.7 TCP communication between SAMS and SIMA . . . 25
3.8 Initial conditions . . . 26
3.9 Intermittent output of timestep results . . . 26
3.10 Iterations . . . 27
3.11 Simulation setup . . . 27
3.11.1 Modifications of the CSC 10 MW model . . . 27
3.11.2 Ice field properties . . . 29
3.11.3 Global parameters of the simulation in SAMS . . . 29
3.11.4 Environmental properties and simulation frequency . . . 30

4 Results **33**
4.1 Effectiveness of coupling . . . 33
4.1.1 Effectiveness of coupling during uninterrupted ice contact . . . 36
4.2 Influence of simulation frequency . . . 37
4.2.1 Surge . . . 39
4.2.2 Sway . . . 40
4.2.3 Heave . . . 41
4.2.4 Roll . . . 42
4.2.5 Pitch . . . 43
4.2.6 Yaw . . . 45
4.3 Influence of ice loads in comparison to other loads on the wind turbine . . . 46
4.3.1 Calm sea . . . 48
4.3.2 Constant wind . . . 49
4.3.3 Turbulent wind . . . 51
4.3.4 Waves . . . 52
4.3.5 Waves and constant wind . . . 53

5 Discussion **55**

5.1 Timestep sensitivity 55
5.2 Ice forces 55
5.3 Cone design 55

6 Conclusion **57**

6.1 Future work 58

Bibliography **59**

Appendix **63**

A Inconsistency with TCP timing in SAMS **64**

B Text output from SAMS **65**

C Treatment of timestamps in the information exchange **67**

D Information available in SIMA **68**

E Global parameters of the simulation in SIMA **70**

List of Figures

1 World wind energy generation by region, required for 2050 net zero 1

2 Mean offshore power density in the North Atlantic at a height of 150 m from sea level 2

3 Global floating substructure market share 3

4 Limits to ice loads 4

5 Failure of ice driven against a sloped structure 5

6 Description of ice contact model 6

7 Competition between different failure modes for a square ice floe of varying sizes and thicknesses 7

8 SIMO-RIFLEX-AeroDyn overview 16

9 Illustration of different modules within SAMS 17

10 Description of the structural model and external load model of the semi-submersible floating wind turbine 18

11 Dimensions of the ice-breaking cones 19

12 Meshing of the semi-submersible geometry 20

13 Difference between global coordinate frames in SIMA and SAMS 20

14 High level overview of the SAMS simulation environment together with the external software SIMA 25

15 Configuration of the external DLL force in SIMA 28

16 Floater in front of ice field 29

17 Turbulent wind speed in the global X direction 31

18 Wave elevation history generated in SIMA 32

19 Coupling agreement in the SIMA global X direction 33

20 Coupling agreement in the SIMA global Y direction 34

21 Coupling agreement in the SIMA global Z direction 34

22 Coupling agreement in the SIMA joint ϕ direction 35

23 Coupling agreement in the SIMA joint θ direction 35

24 Coupling agreement in the SIMA joint ψ direction 36

25 Ice force on the floater in the SIMA global X direction 36

26 Coupling agreement in the SIMA global X direction 37

27 Final state of simulation with 5 Hz 38

28 Final state of simulation with 10 Hz 38

29 Final state of simulation with 20 Hz 39

30 Ice forces on the floater in the X direction 39

31 Displacement of the floater in the X direction 40

32 Ice forces on the floater in the Y direction 41

33 Displacement of the floater in the Y direction 41

34 Ice forces on the floater in the Z direction 42

35 Displacement of the floater in the Z direction 43

36 Ice forces on the floater in the ϕ direction 43

37 Displacement of the floater in the ϕ direction 44

38 Ice forces on the floater in the θ direction 45

39 Displacement of the floater in the θ direction 45

40 Ice forces on the floater in the ψ direction 46

41 Displacement of the floater in the ψ direction 46

42 Longitudinal ice force on the structure in calm sea 48

43 Longitudinal displacement of the structure in calm sea 48

44 Longitudinal ice force on the structure with constant wind 49

45 Submergence and emergence of ice cones due to pitch 49

46 Incoming longitudinal wind speed on the turbine shaft in constant wind . . 50

47 Longitudinal displacement of the structure with constant wind 50

48 Longitudinal ice force on the structure with turbulent wind 51

49 Incoming longitudinal wind speed on the turbine shaft in turbulent wind . . 51

50 Longitudinal displacement of the structure with turbulent wind 52

51 Longitudinal ice force on the structure in sea waves 52

52 Longitudinal displacement of the structure in sea waves 53

53 Longitudinal ice force on the structure with sea waves and constant wind . 53

54 Incoming longitudinal wind speed on the turbine shaft in sea waves and constant wind 54

55 Longitudinal displacement of the structure with sea waves and constant wind 54

A.1 Example of a 5 Hz simulation ending with a timing error 64

A.2 Example of a 20 Hz simulation ending without a timing error 65

D.1 Input parameters to the coupling code 69

List of Tables

1 Different methods in the Newmark-family 12

2 Hydrostatic restoring coefficients constituting the stiffness matrix **K** of the floater 18

3 Mass coefficients constituting the mass matrix **M** of the floater 19

4 Elements with mass in the SIMA model of the floating wind turbine 26

5 Statistical properties of the ice field and mechanical properties of ice 30

6 Simulation conditions for different simulated scenarios 31

7 Time spent simulating one hour of a coupled scenario with wind and waves 37

8 Statistical summary of ice forces in the X direction 40

9 Statistical summary of floater displacement in the X direction 40

10 Statistical summary of ice forces in the Y direction 41

11 Statistical summary of floater displacement in the Y direction 42

12 Statistical summary of ice forces in the Z direction 42

13 Statistical summary of floater displacement in the Z direction 43

14 Statistical summary of ice torques around the X axis 44

17 Statistical summary of floater displacement in the θ direction 44

15 Statistical summary of floater displacement in the ϕ direction 44

16 Statistical summary of ice torques around the Y axis 45

18 Statistical summary of ice torques around the Z axis 46

19 Statistical summary of floater displacement in the ψ direction 47

20 Statistical summary of longitudinal ice forces on a floater 47

21 Statistical summary of longitudinal displacements of a floater 47

A.1 Inconsistency with TCP timing in SAMS 64

D.1 Information passed to the coupling code from SIMA 68

List of abbreviations

Latin Symbols

A	Ice floe area
A_{rotor}	Area swept by wind turbine rotor
c	Cohesion of ice rubble
$C_{p,max}$	Maximum wind power extraction coefficient
C_1	Linear current force coefficient
C_2	Quadratic current force coefficient
D	Ice piece width
$h^{(1)}$	Linear impulse response function
$h^{(2)}$	Second-order impulse response function
H	Weight function on a cracked body
F_{S1}	Force on structure from large ice pieces
F_{S2}	Force on structure from loose ice rubble
$F_{S3total}$	Force on structure from pressured ice rubble
F_Y	Lateral load on ice
$F_{Z,B}$	Breakthrough load causing the ice floe to fail
$F_{Z,C0}$	Maximum load required to initiate circumferential ice crack
$F_{Z,R0}$	Maximum load required to initiate radial ice crack
$F_{Z,R1}$	Variable load required to propagate radial ice crack
k	Time step index
K	Bearing capacity factor
$\mathbf{K}$	Hydrostatic stiffness matrix
K_0	Pressure ratio of loose ice rubble
K_{IC}	Fracture toughness of ice
m	Meter
$\dot{m}$	Mass flux of air through wind turbine rotor
p	Pressure
q	Average shear strength of ice rubble
q_{CU}	Drag force component
$\mathbf{q}_{CU}$	Drag force vector
$\mathbf{q}_{hydrostatic}$	Hydrostatic force vector
$\mathbf{q}_{WA}^{(1)}$	First-order wave excitation force vector
$\mathbf{q}_{WA}^{(2)}$	Second-order wave excitation force vector
Pa	Pascal
p_{ext}	External stress profile of ice crack
P_{max}	Maximum amount of wind energy possible to extract
t	Time
t_{ice}	Ice thickness
$\bar{u}$	Average wind velocity at given height
u	Relative velocity between body and current
z	Vertical distance
z_r	Reference vertical distance
V_0	Wind speed
v_1	Current velocity component
v_2	Current velocity component
$\mathbf{x}$	Body displacement vector
$\mathbf{x}_0$	Hydrostatic stiffness reference vector
$\dot{x}_1$	Body velocity component

$\dot{x}_2$	Body velocity component

Greek symbols

α_{CU}	Relative angle between body and current
α_{wind}	Height coefficient
γ	Buoyant weight of ice rubble
Δt	Time-step duration
μ	Ice-structure friction coefficient
ρ_a	Density of air
ρ_i	Density of ice
ρ_w	Density of water
ϕ	Rotational displacements around the local X" axis
ϕ_{ice}	Friction angle of ice rubble
ψ	Rotational displacements around the local Z" axis
χ	Position along ice thickness
θ	Rotational displacements around the local Y" axis

Abbreviations

2D	Two-dimensional
BEM	Blade element / momentum
CO_2	Carbon dioxide
DOF	Degree of freedom
DEM	Discrete element method
DLL	Dynamically linked library
f	Simulation frequency
GUI	Graphical user interface
GPU	Graphics processing unit
Hz	Hertz
JONSWAP	Joint North Sea Wave Project
kg	Kilogram
LEFM	Linear elastic fracture mechanics
MCD	Mean caliper diameter
MW	Megawatts
NDEM	Non-smooth discrete element method
NPD	Norwegian Petroleum Directorate
PIC	Particle-in-Cell approach
PV	Photovoltaics
SAMS	Simulator for Arctic Marine Structures
SI	International system of measurements
TCP	Transmission control protocol

1 Introduction

”Net zero” is a goal of zero global carbon dioxide (CO_2) emissions from energy, processes and land use by the year 2050. Reaching it is extremely challenging (DNV, 2021a). Despite their best efforts, many developing nations and hard-to-abate sectors are very unlikely to achieve zero emissions by 2050. For developed nations and easy-to-electrify sectors, this means that they must reduce their carbon footprint past zero and become carbon-negative. Moreover, they will have to achieve it before 2050.

In the scenario that global warming is successfully limited to 1.5°, electricity will have to account for 51 % of total global energy demand. Of that, 86 % must be supplied by renewable sources. These, in turn, must be globally dominated by wind, as it is expected to eclipse solar photovoltaics (PV) as the largest energy source by the year 2046 (DNV, 2021a). A large increase in installed wind capacity is therefore required, as shown in Figure 1.

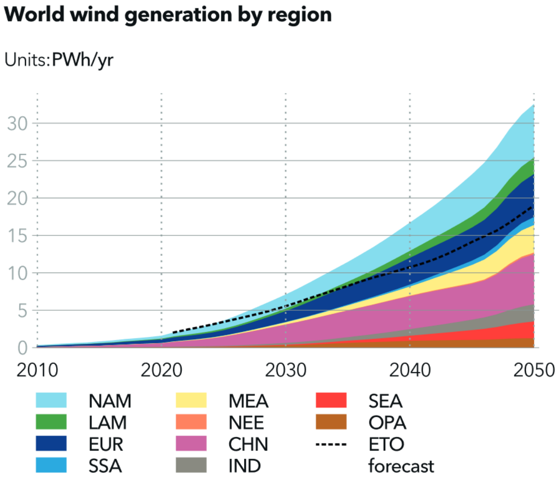


Figure 1: World wind energy generation by region, required for 2050 net zero.

Source: DNV, 2021a; Global CCS Institute, 2021; IEA, 2020

1.1 Floating offshore wind turbines

Size has been found to be one of the main driving factors in increasing wind turbine effectiveness (Musial et al., 2021). It is preferable to use the largest available wind turbine for a given project, because the same plant capacity can be achieved with fewer turbines (Shields et al., 2021). Offshore wind turbines face less transportation constraints than land-based wind turbines and can therefore be larger (Bilgili et al., 2011).

Floating wind turbines give access to at least four times as much ocean surface space, compared with bottom-fixed installations (DNV, 2021b). This provides increased choice in site selection, allowing to utilize areas with more favourable wind conditions. Additionally, the increase in available space leads to lower environmental and social impact of large-scale wind installations. The main drawback of floating wind turbines, cost, is rapidly losing significance. For example, investment cost per MW for Hywind Tampen, installed juts recently, is 40 % lower than Hywind Scotland, installed in 2017 (Equinor, 2021).

The advantage of increased available installation area can be further improved. As shown in Figure 2, there are large areas of the ocean in the northern latitudes where mean wind power density is relatively high.

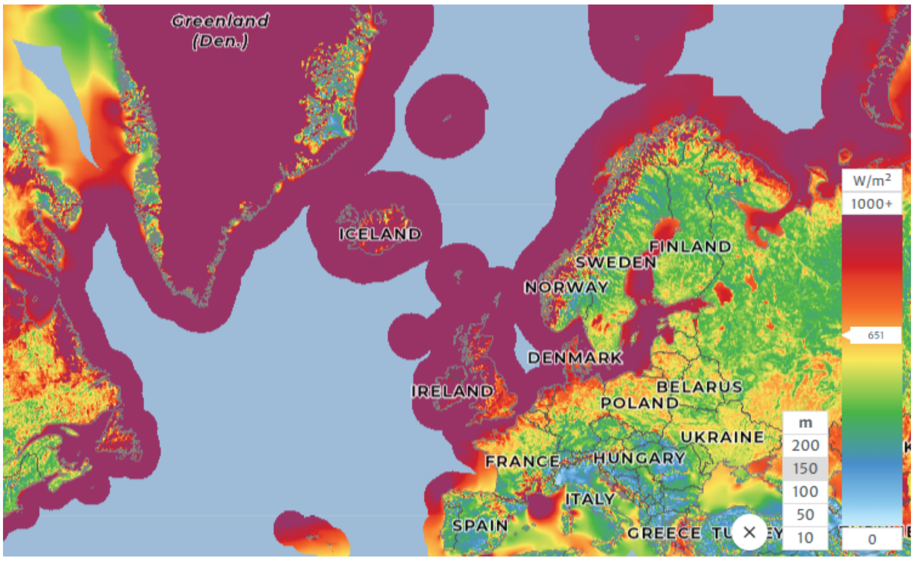


Figure 2: Mean offshore power density in the North Atlantic at a height of 150 m from sea level

Source: DTU, 2021

However, the existence of areas rich in wind energy does not mean they are immediately accessible for development. Installing wind capacity in cold regions carries several challenges, not the least of which is drifting sea ice. Any structure designed for these hostile regions must be designed with drifting sea ice in mind (Liferov, 2014). The aim of this thesis is to improve the existing range of analysis tools to enable simulating floating wind turbines in drifting sea ice conditions. With this contribution, another piece of the groundwork will be laid that helps accelerate our pathway to net zero.

2 Literature review and theory

In this chapter, a brief review is provided covering aspects of the simulation of this thesis. Semi-submersible floating wind platforms are introduced. Descriptions are given for forces that are expected to act on such a platform in a cold climate sea environment. A literature summary is made about the methods used to find ice forces, especially in the time domain. Different ways are presented to describe an orientation of a structure in space and time, as well as conversions between them.

2.1 Semi-submersible floating wind platforms

There are currently more than 40 concepts of floating wind turbines under development (DNV, 2021b), with new concepts being frequently announced. To date, their relative adoption seems to favour semi-submersible platforms, as apparent from Figure 3.

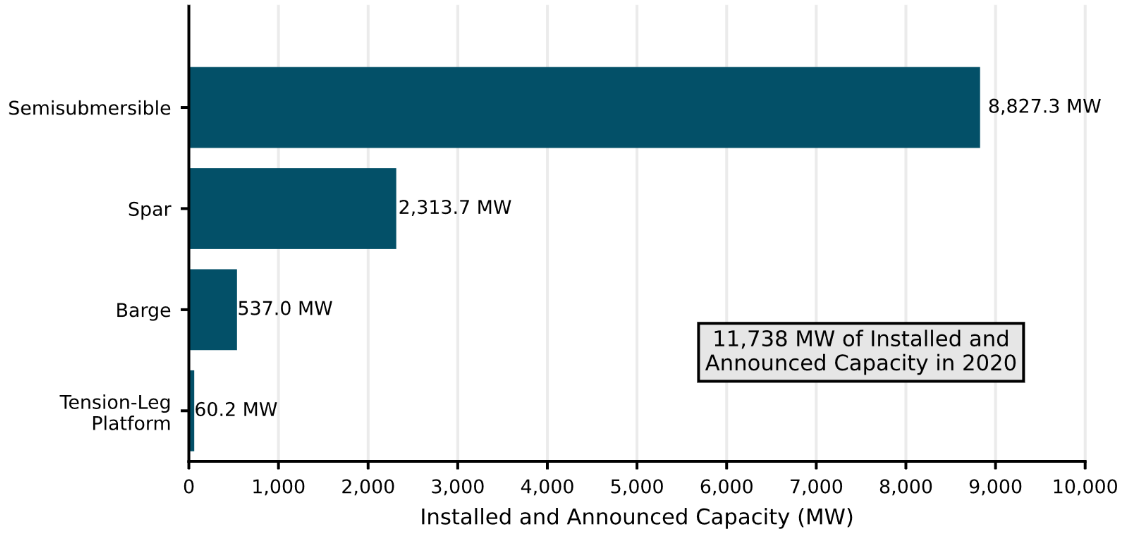


Figure 3: Global floating substructure market share.

Source: Musial et al., 2021

An example of a semi-submersible platform is illustrated for example in Figure 13. Semi-submersibles are stabilized by their columns, which are connected by pontoons below the waterline. Compared to spar platforms, they have a lower draft and therefore present an easier challenge in transport and installation. The mooring and anchoring systems of semi-submersibles are simple. Their seakeeping stability is ensured by the waterplane area moment of inertia, so failure of the mooring system does not imply a fatal event for the structure itself. (Pettineo, 2021)

As a main drawback, the floater can experience large heave motions that present problems for the power export cable. (Wang, 2014).

2.2 Ice forces on structures

As it pushes against structures, moving ice can damage or destroy them. In Figure 4, Palmer and Croasdale, 2012 identify several factors that can set an upper bound on the exerted global force.

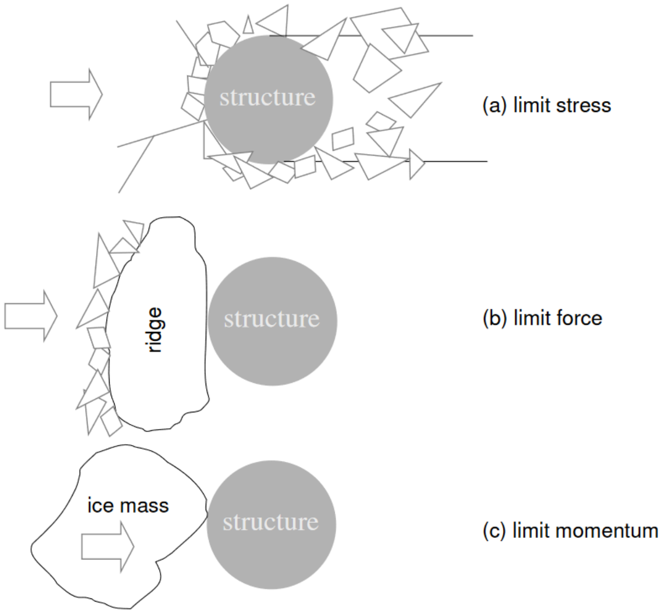


Figure 4: Limits to ice loads.

Source: Palmer and Croasdale, 2012

Starting from the bottom of the illustration, an ice mass may have low enough momentum that it is absorbed in the structure, stopping the ice mass from moving any further and exerting any further force. A floating or moored structure would then change its velocity due the gained momentum. For larger ice floes, the force on the structure is limited by the driving forces of the ice sheet - wind, current, and other ice floes behind the one that is in contact with the structure. If the ice floe has enough momentum and enough force driving it against the structure, the next and highest limit is the stress limit. It is determined by the strength of the ice. If limit stress is reached, the ice fails against the structure.

There are several ways for the ice to fail. Palmer and Croasdale, 2012 identifies four main failure mechanisms. If the ice movement is very slow, the ice fails in creep. If it is thin, it can fail in buckling. A common, and often considered worst-case, failure mode is crushing. Finally, in the presence of vertical forces, the ice can fail in bending. Section 2.2.1 elaborates on ice floes failing in fracture.

When designing structures for ice loads, the goal is often to prevent ice from failing in crushing and to make sure it fails in bending. Generally, bending failure happens at much lower driving forces than crushing, and thus is more favourable for the structure. The bending of level ice against a sloping structure is illustrated in Figure 5.

Here, the unbroken ice (1) in Figure 5 is driven from left to right towards a static structure. Being pushed onto the slope, it is lifted out of the water (2). Once the bending moment from its own weight exceeds its flexural strength, the ice breaks. In case of downward-sloping structures, such as the cones proposed in Section 3.3.2, it is the buoyancy of the

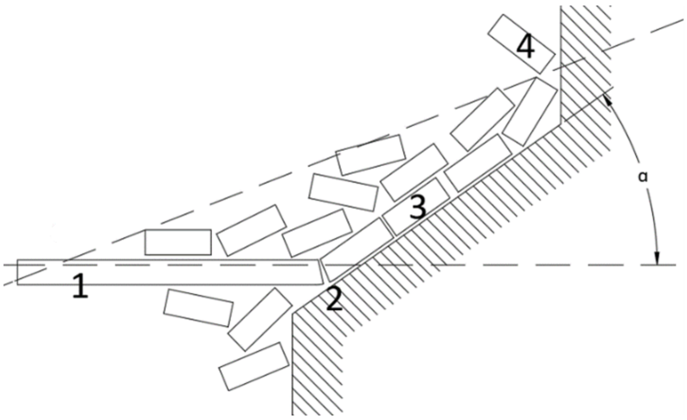


Figure 5: Failure of ice driven against a sloped structure.

ice that causes it to fail in bending. The broken-off pieces are pushed along the sloped surface of the structure (3) by the ice behind them. Upon reaching the vertical part of the structure, the ice piece is overturned (4). It either joins the accumulating pile of ice rubble, or floats away in case of a narrow structure.

2.2.1 Fracture of ice floes

The different modes of ice fracture are further explored in Lu et al., 2016. In momentum-limited interactions, a floe of finite size will simply change course and rotate. The structure, if it is free to move, will do the same. In the case of force-limited interactions, the ice floe can fail either in flexural failure out-of-plane (Lu et al., 2015b) or it can split in-plane (Lu et al., 2015a).

Lu et al., 2015b mention two different out-of-plane failure types. Finite floes can fail in fracture controlled by radial crack initiation. Semi-infinite floes can fail in sequential forming of radial and circumferential cracks. It is preferable that a floe fails in radial-crack-initiation-controlled failure, because it leads to lower lower ice forces on the structure. The interaction of a structure with the ice floe edge is modeled by considering an isolated loading area. The loads on that area in three different directions are illustrated by Lu et al., 2015b.

For a semi-infinite floe, the vertical force load F_Z exerted by the structure on the ice will reach several important values, as it grows.

- $F_{Z,R0}$: Maximum load required to initiate the radial crack
- $F_{Z,R1}$: Variable load required to propagate a radial crack, depending on the length of the radial crack
- $F_{Z,C0}$: Maximum load required to initiate the circumferential crack
- $F_{Z,B}$: Final breakthrough load, causing the ice floe to fail in out-of-plane bending.

In the case of smaller ice floes, the final breakthrough load $F_{Z,B}$ will be reached on first contact and the circumferential cracks will not form. An example of these different failure mechanisms can be observed in Figure 45. The left-most column of the semi-submersible

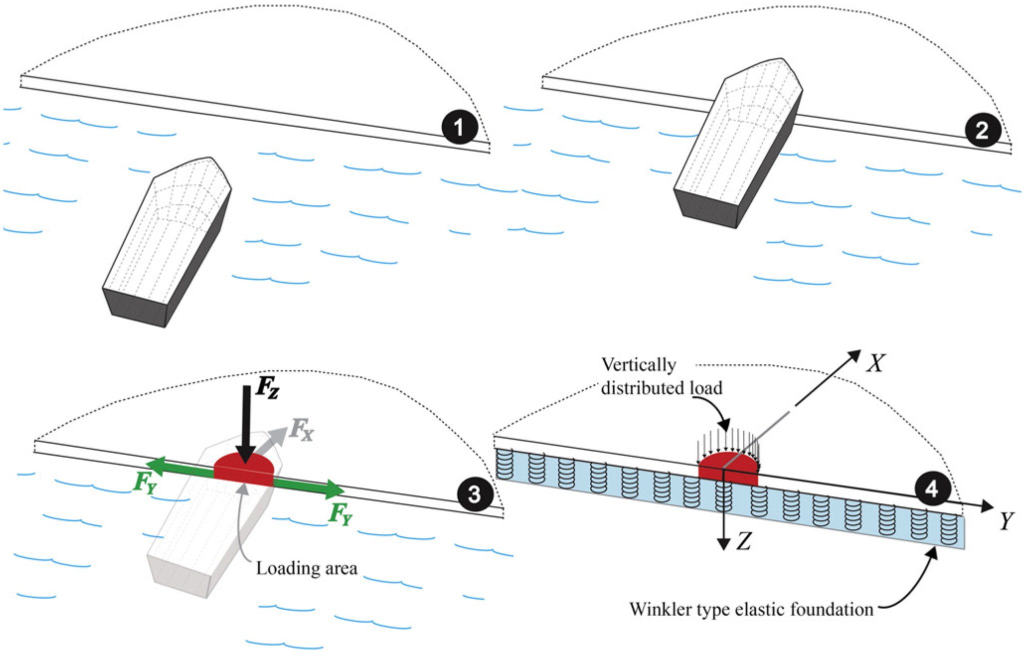


Figure 6: Description of ice contact model.

Source: Lu et al., 2015b

floater has caused radial cracks visible under the ice-breaking cone. They end at a circumferential crack. As a result, some triangular ice pieces have separated from the ice floe and are floating away. Finally, one of the radial cracks has propagated all the way to the opposite end of the floe, at the left edge of the figure.

On Figure 6, the lateral force pair F_Y can also cause an in-plane split of the floe. If that happens, any continuous local bending failure will cease to develop. Splitting failure can thus act as a mechanism of load release. Lu et al., 2015a find that most ice floes with sizes of engineering interest can be described with the help of the linear elastic fracture mechanics (LEFM) approach, aided with the weight function method. The lateral load $F_Y(A)$ necessary for ice splitting is found to depend on the fracture toughness K_{IC}, an external stress profile $p_{ext}(\chi)$, and a weight function $H(A, \chi)$ of a cracked body.

$$\frac{F_Y(A)}{t} = \frac{1}{H(A,0)} \left[K_{IC} + \int_0^A p_{ext}(\chi) H(A,\chi) \mathrm{d}\chi \right]$$

Lu et al., 2016 propose that any of these failure mechanisms will alleviate the load, when they occur. Thus they prevent the other mechanisms from happening, resulting in a kind of competition. This competition is influenced by the size of the floe, as Lu et al., 2016 illustrate in Figure 7.

A small ice floe up to 8 m will impart some of its momentum to the structure, change course, and rotate. A larger floe of approximately 10 m may fail with one radial crack due to vertical force F_Z. Floes between 20 m and 50 m are expected to split in-plane due to lateral forces F_Y. Above that size, the ice floe may be considered semi-infinite and will fail with successive radial and circumferential cracks.

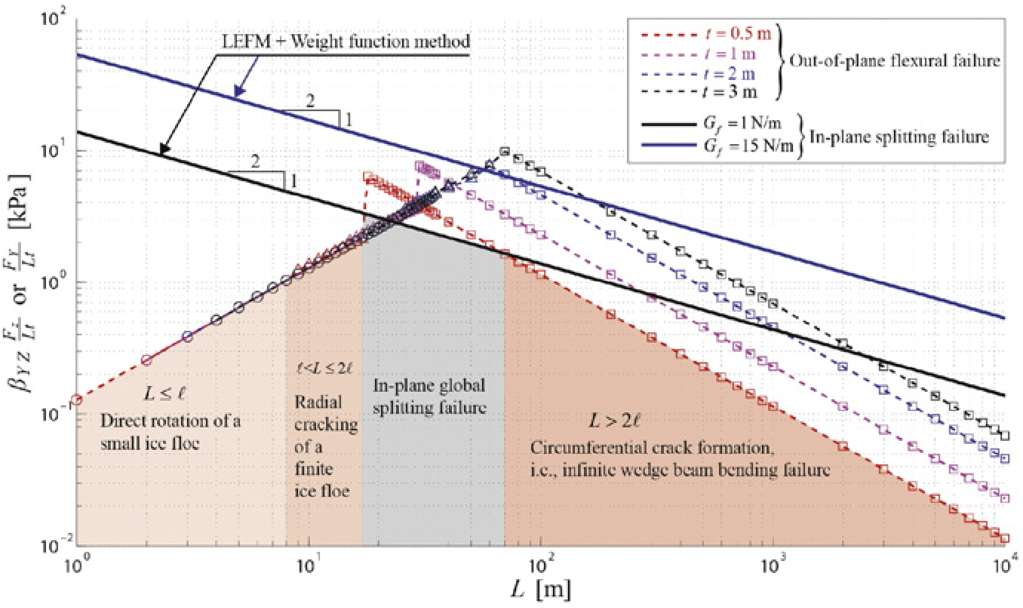


Figure 7: Competition between different failure modes for a square ice floe of varying sizes and thicknesses.

Source: Lu et al., 2016

2.2.2 Broken ice

Croasdale et al., 2009 classify interactions between broken ice and a structure into three scenarios. In the first scenario, the thick ice features have broken into large pieces that collide with the structure. In this case, the force F_{S1} exerted on the structure is estimated to depend on the ice piece width D, ice thickness t_{ice} and pressure p imposed by surrounding broken ice.

$$F_{S1} = ptD$$

$$p = 2000 t_{ice}^{0.25} D^{-0.54} [\text{kPa}]$$

In the second scenario, the ice is broken into small pieces, loosely packed, and experiences slip plane failure. The load on the structure F_{S2} is given by Croasdale et al., 2009.

$$F_{S2} = KqDt_{ice}$$

$$q = \frac{K_0 \gamma_{eff} \tan \phi_{ice}}{2} + c$$

$$\gamma_{eff} = (1 - e)(\rho_w - \rho_i)$$

Here, $K = 6$ is a bearing capacity factor reflecting the length of the slip planes compared to the structure width. q is the average shear strength of the ice rubble along the slip planes. γ_{eff} is the buoyant weight of the ice rubble. $K_0 = 1$ is the ratio of vertical to lateral pressure of loose ice rubble. ϕ_{ice} and c are the ice rubble friction angle and cohesion, respectively. ρ_w and ρ_i are respective densities of water and ice.

In the third scenario, all ice is broken into small ice pieces and pressured ice is possible. Some ice will collect into a stationary wedge in front of the structure, with an angle a

between the wedge side and ice drift direction. The ice load $F_{S3total}$ on this wedge can be approximated as:

$$F_{S3total} = pDt_{ice}\left(1 + \frac{\mu}{\tan a}\right) + 2pDt_{ice}\mu$$

Here, μ is the ice-structure friction coefficient.

2.3 Time-domain methods for calculating broken ice loads on moored structures

Closed-form methods such as the one described in Section 2.2.2 suffer from lack of versatility. To increase the range of scenarios where a method is applicable, time-domain simulation is suitable. There is a review written by Islam et al., 2021 that focuses on analysis methods for interactions between broken ice and marine structures. Another review, focusing specifically on the discrete element method (DEM), was done earlier in Tuhkuri and Polojärvi, 2018.

Here, the review will be further narrowed down to include only methods that support studying moored floating structures.

Murray and Spencer, 1997 described DECICE, a 2D simulation tool. They used it to simulate a turret moored tanker in pack ice. The mooring was modeled as a nonlinear force depending on displacement. Lau, 2006 performed a simulation of an ice-breaking ship with the same software. More simulations were performed in Lau et al., 2011.

E. Hansen and Løset, 1999b used a DEM model to simulate a moored vessel in an advancing broken ice field. This simulation was compared to the actual tank test in E. Hansen and Løset, 1999a. This model is missing fracture and cracking of ice floes.

Barker et al., 2000 used the Particle-in-Cell approach (PIC) to model the Kulluk ice platform in pack ice conditions. The simulation was backed up by field data from the Kulluk. The simulation was two-dimensional. The Kulluk was moored and thruster-assisted. This method was developed further in Sayed and Barker, 2011 and Sayed et al., 2012. It was applied to simulate a moored and thruster-assisted drillship in Sayed et al., 2014, Sayed et al., 2015 and Sayed et al., 2016. It was applied to find acceptable ice conditions for a vessel in Sayed et al., 2018.

Karulin and Karulina, 2010 used a DEM model to simulate a vessel moored to a production platform in a moving ice field. The simulation was performed in two translational degrees of freedom. The model was validated with ice tank tests in Karulin and Karulina, 2011. The model was also applied to find mooring loads on a multi-legged semi-submersible in Karulin and Karulina, 2013. More ice tank tests were performed to study the same multi-legged semi-submersible in Karulin and Karulina, 2014.

Eik, 2011 provided a structural reliability analysis method that incorporates managed ice loads in a larger framework. There is no simulation involved, but some closed-form equations. This is a continuation of Eik and Aksnes, 2010, where a relation is given for peak and mean mooring loads.

Lubbad and Løset, 2011 used the PhysX engine combined with closed-form solutions for ice fracture to perform simulations that are fast enough for real-time analysis. This was later expanded into Lubbad et al., 2018. A non-smooth time-stepping scheme, described in van den Berg et al., 2018, was incorporated in the software. Examples of its use were provided in Tsarau et al., 2018. It was used to study effects of ice floe shape in van den

Berg et al., 2019. Further developments to the software were described in Lubbad et al., 2020. The equations of motion used in this software are further elaborated in Section 3.2.

Amdahl et al., 2014 used the NDEM method to investigate a moored drillship. The goal was to simulate the ship's response in real time. The ice field in this study was modeled as managed ice, meaning an assisting ice-breaking ship broke the ice into smaller pieces before it reached the drillship.

Bonnemaire et al., 2014 used SimShipIce together with ice basin tests, with a goal of studying moored structures such as the Cat-I moored drillship in Bonnemaire et al., 2015. By that time, the software was named SIBIS and developed together with Statoil and Multiconsult.

Richard and McKenna, 2013 describes a DEM model in 2D used to simulate a turret-moored ship in pack ice.

Metrikin, 2014 described what was then called the "Numerical ice tank" with a claim for dynamic positioning and mooring support. This was then developed into Metrikin, Gürtner et al., 2015. Ice tank validations were done in Metrikin, Teigen et al., 2015. Further improvement and validation are described in Nicolas et al., 2019.

Liu et al., 2016 used a DEM method, accelerated by a graphical processing unit (GPU), to simulate the Kulluk platform with linear mooring. The ice was level ice.

Su et al., 2019 used a tool called FhSim to simulate the moored Kulluk platform in pack ice. FhSim is developed by SINTEF Ocean in Trondheim, Norway.

A floating wind turbine is an elastic structure, fixed with nonlinear catenary mooring to the seabed, subject to wind and waves, with blade pitch controlled by a numerical controller. Of all the methods mentioned above, there is not a single one that allows for analysis of all these effects, together with broken ice drifting on the sea surface.

2.4 Other forces acting on a semi-submersible wind turbine

2.4.1 Wind thrust

As the wind passes through the area A_{rotor} swept by the wind turbine blades, it is slowed down and its kinetic energy is transformed into mechanical energy at the shaft. For a wind speed V_0, the maximum amount of energy P_{max} could be extracted if it was possible to slow the wind down to a halt.

$$P_{max} = \frac{1}{2}\dot{m}V_0^2 = \frac{1}{2}\rho_a A_{rotor} V_0^3$$

Here, ρ_a is the density of the air. Reducing the wind speed to zero is not, in fact, possible. M. Hansen, 2015 describes an inherent limitation of wind power available for extraction, called the Betz limit $C_{p,max}$.

$$C_{p,max} = \frac{16}{27} \approx 0.593$$

It is therefore not possible to extract more than 59.3 % of the maximum available wind power at the wind turbine. In the context of this thesis, the wind thrust is calculated through aerodynamic forces acting on the wind turbine blades. For this purpose, the blade element momentum (BEM) model is used. The aerodynamic forces are applied to the wind turbine tower modeled as a system of slender elements. The slender element system then applies a force to the semi-submersible rigid body. This force is included in the external force $\mathbf{q}_{add_{SIMA}}$ in Section 3.4.

The wind itself is described in SINTEF, 2018 as propagating parallel to the horizontal plane. Wind gusts are modeled as a Gaussian stochastic process. The average wind velocity $\bar{u}$ at a height z from sea-level is given in equation 3.1 of SINTEF, 2018.

$$\bar{u}(z) = \bar{u}_r \left(\frac{z}{z_r}\right)^{\alpha}$$

Here, z_r is a reference height, typically 10 m. α is a height coefficient, typically chosen from a range between 0.10 and 0.14. The time-varying part is given by a spectrum proposed by the Norwegian Petroleum Directorate (NPD). The NPD spectrum described in section 3.1.7 of SINTEF, 2018.

2.4.2 Wind and current drag force

The wind exerts a drag force $\mathbf{q}_{WI}$ on the turbine tower and the sea current exerts a drag force $\mathbf{q}_{CU}$ on the catenary mooring system. These drag forces are described in SINTEF, 2018 with the so-called Morrison equation.

$$\begin{aligned} q_{CU}^{(DOF)}(\alpha_{CU}, t) &= C_1^{(DOF)}(\alpha_{CU})|u(t)| + C_2^{(DOF)}(\alpha_{CU})|u(t)|^2 \\ |u|^2 &= (v_1 - \dot{x}_1)^2 + (v_2 - \dot{x}_2)^2 \\ \alpha_{CU} &= \arctan \frac{v_2 - \dot{x}_2}{v_1 - \dot{x}_1} \end{aligned}$$

Here, DOF is the degree of freedom. C_1 and C_2 are respectively the linear and quadratic current force coefficients. u is the relative velocity between low frequency body velocity and current velocity. α_{CU} is the relative angle between direction of low frequency body velocity and current velocity. v_1 and v_2 are the current velocity components at the surface for floating bodies, such as the semi-submersible, and at the centre of gravity for submerged bodies, such as the turbine tower. $\dot{x}_1$ and $\dot{x}_2$ are the body velocity components in the body coordinate system.

2.4.3 Wave excitation forces

SINTEF, 2018 splits the wave excitation forces into three contributions. The first-order forces $\mathbf{q}_{WA}^{(1)}$ oscillate with wave frequencies. The second-order forces $\mathbf{q}_{WA}^{(2)}$ include mean, rapidly and slowly-varying wave drift forces. They are calculated according to equation 4.29 of SINTEF, 2018:

$$\mathbf{q}_{WA}^{(1)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} h^{(1)}(\tau_1)\zeta(t - \tau_1)\mathrm{d}\tau_1$$

$$\mathbf{q}_{WA}^{(2)} = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h^{(2)}(\tau_1, \tau_2)\zeta(t - \tau_2)\zeta(t - \tau_1)\mathrm{d}\tau_1\mathrm{d}\tau_2$$

Here, $h^{(1)}$ is the linear impulse response function and $h^{(2)}$ is the second-order impulse response function.

2.4.4 Buoyancy forces

In this thesis, the buoyancy force acting on a floating body is approximated as a hydrostatic stiffness force. It is given in SINTEF, 2018 as a function of the hydrostatic stiffness matrix $\mathbf{K}$, body displacement $\mathbf{x}$ and stiffness reference position $\mathbf{x}_0$.

$$\mathbf{q}_{hydrostatic} = \mathbf{K}(\mathbf{x} - \mathbf{x}_0)$$

2.5 Newmark-beta time-stepping scheme

When an analytical solution is impractical or won't capture all desired phenomena, time domain analysis is used. The Nemark-Beta method is utilized in SIMA. SAMS uses its own time-stepping scheme described in van den Berg et al., 2018, but it is still derived from the Newmark-Beta method.

SINTEF, 2018 refers to the position vector $\mathbf{x}$, velocity vector $\dot{\mathbf{x}}$, and acceleration vector $\ddot{\mathbf{x}}$ of a rigid body.

$$\begin{aligned} \mathbf{x} &= (x, y, z, \phi, \theta, \psi)^T \\ \dot{\mathbf{x}} &= (\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})^T \\ \dot{\mathbf{x}} &= (\ddot{x}, \ddot{y}, \ddot{z}, \ddot{\phi}, \ddot{\theta}, \ddot{\psi})^T \end{aligned}$$

The properties of the sub-vectors containing rotations ϕ, θ, ψ are discussed in Section 2.6. The dynamic model is expressed by a function f with an input ξ, which contains everything not depending on ($\mathbf{x}$ or $\dot{\mathbf{x}}$, such as wind or current.

$$\ddot{\mathbf{x}} = f(\mathbf{x}, \dot{\mathbf{x}}, \xi)$$

Assuming a timestep k with time t_k and timestep duration Δt, a following short-hand notation is adapted from SINTEF, 2018:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{x}(t_k) \\ \dot{\mathbf{x}}_k &= \dot{\mathbf{x}}(t_k) \\ \xi_k &= \xi(t_k) \\ f_i &= f(\mathbf{x}_k, \dot{\mathbf{x}}_k, \xi_k) \end{aligned}$$

The method employs a predictor described in equation 6.37 of SINTEF, 2018.

$$\begin{aligned} \dot{\mathbf{x}}_{k+1}^{(0)} &= \dot{\mathbf{x}}_k + \Delta t \cdot f_k \\ \mathbf{x}_{k+1}^{(0)} &= \mathbf{x}_k + \Delta t \cdot \dot{\mathbf{x}}_{k+1}^{(0)} \end{aligned}$$

After that, a corrector is applied for a preset amount of times.

$$f_{k+1}^{(i)} = f\left(\mathbf{x}_{k+1}^{(i)}, \dot{\mathbf{x}}_{k+1}^{(i)}, \xi_{k+1}\right)$$
$$\dot{\mathbf{x}}_{k+1}^{(i+1)} = \mathbf{dotx}_k + T(1-\gamma)f_k + \gamma f_{k+1}^{(i)}$$
$$\mathbf{x}_{k+1}^{(i+1)} = \mathbf{x}_k + T\dot{\mathbf{x}}_k + (\frac{1}{2} - \beta)T^2 f_k + \beta T^2 f_{k+1}^{(i)}$$

As long as $\gamma = \frac{1}{2}$, there will be no artificial damping introduced by the time integration scheme. Then, the choice of β depends on the stability of the method. The method may become unstable if the chosen timestep Δt is larger than a critical value h_{cr}, which in turn depends on the natural period of the system T. Langen and Sigbjörnsson, 1979 identify special cases of the Newmark scheme. Depending on the choice of β, these special cases reduce to other time integration schemes described by previous authors.

β	Description	Stability condition
0	Second central difference	$h_{cr} < 0.318T$
$\frac{1}{12}$	Fox-Goodwin's method	$h_{cr} < 0.389T$
$\frac{1}{6}$	Linear acceleration	$h_{cr} < 0.551T$
$\frac{1}{4}$	Constant average acceleration	Unconditionally stable

Table 1: Different methods in the Newmark-family

2.6 Representations of orientation in 3D space

In this section, several equivalent ways are summarised of representing an orientation of a rigid body, with regards to a fixed coordinate frame. The fixed coordinate frame will be referred to as the global frame, while the coordinate frame that moves with the rigid body will be called the local, or body, frame.

2.6.1 Matrix transformation

An easy way to represent a rotation around a fixed point in space is to use a transformation matrix (O'Reilly, 2020). If a vector in the body coordinate system is $(x^B, y^B, z^B)^T$, then the same vector as seen from the global coordinate system is $(x^G, y^G, z^G)^T$. The transformation matrix $\mathbf{R}$ represents a rotation from the global orientation to the local one.

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \mathbf{R} \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix}$$

In this summary $\mathbf{R}$ will be called a rotation matrix, but it does not have to represent only rotations. Based on its elements, a transformation matrix can also represent scaling, skewing, or mirroring. Because of this generality, this will be considered the default way to represent an orientation.

Marintek, 2009 shows that in order to represent a rotation from the local back to the global frame, the inverse $\mathbf{R}^{-1}$ is used. In the case of rotations, the matrix is proper orthogonal and thus its inverse is also its transpose.

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = \mathbf{R}^T \begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix}$$

2.6.2 Joint rotations

Another way to describe an orientation of a structure is to define it through an successive series of rotations around certain axes. O'Reilly, 2020 describes such rotation sets as Euler angles, but also Tait and/or Bryan angles, or the Euler-Cardan angles. Because of the property that the axis of each next rotation depends on the previous one, they are also qualitatively called "joint rotations". This highlights the similarity to a rigid joint system. Twelve different sets of axes could be chosen for these rotations, with any of them making it possible to completely describe any orientation of a rigid body with respect to a global coordinate frame. Marintek, 2009 uses what is called the $3-2-1$, or $Z-Y'-X''$ set. In this thesis, the term "joint rotations" will be used.

Describing the transformation that the joint rotations cause, the structure starts from the same orientation as the global coordinate system. First, it rotates around the global axis Z by an angle ψ. Its own X and Y axes rotate as a result. The vector $\mathbf{x}^B$ as seen from the global coordinate frame would at this moment be

$$\mathbf{x}^G = \mathbf{R}_z \mathbf{x}^B$$

or

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix}$$

Then, the structure rotates around its new Y' axis by an angle θ. Its own Z and X' axes rotate as a result. The prime sign indicates that the axis has moved in a previous rotation.

$$\mathbf{x}^G = \mathbf{R}_z \mathbf{R}_y \mathbf{x}^B$$

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix}$$

Finally, the body rotates around its X'' axis by an angle ϕ, with Y' and Z' axes rotating as a result.

$$\mathbf{x}^G = \mathbf{R}_z \mathbf{R}_y \mathbf{R}_x \mathbf{x}^B$$

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} \tag{1}$$

Having reached the end of the rotations, all three axes have moved from the global orientation X, Y, Z to the local orientation X'', Y'', Z''. Multiplying together the matrices of the individual rotations in Equation 1, one can obtain a rotation matrix γ(Marintek, 2009) that represents the whole transformation as a function of angles ϕ, θ, ψ.

$$\mathbf{x}^G = \gamma \mathbf{x}^B$$

$$\begin{bmatrix} x^G \\ y^G \\ z^G \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix}$$

In the opposite case of having such a matrix as input, one can express the rotation angles ϕ, θ, ψ in any number of ways, for example:

$$\theta = \arcsin(-\gamma_{3,1})$$
$$\phi = \arcsin\left(\frac{\gamma_{3,2}}{\cos\theta}\right)$$
$$\psi = \arcsin\left(\frac{\gamma_{2,1}}{\cos\theta}\right)$$

It should be noted that while the joint rotation angles ϕ, θ, ψ represent three degree of freedoms, they do not constitute a vector in a physical space. Rather, they may be considered to represent a vector in a so-called "configuration space", examined further in Section 2.6.4.

2.6.3 Axis-angle representation

O'Reilly, 2020 discusses Leonhard Euler's proposed way of defining any rotation by a vector $\mathbf{e}$ and an angle θ around the axis represented by $\mathbf{e}$. Knowing the axis and angle, one can use Rodrigues' rotation formula (Rodrigues, 1840) to express an rotation matrix from them.

$$\gamma = (1 - \cos\theta) \begin{bmatrix} e_1^2 + \cos\theta & e_1e_2 - e_3\sin\theta & e_1e_3 + e_2\sin\theta \\ e_2e_1 + e_3\sin\theta & e_2^2 + \cos\theta & e_2e_3 - e_1\sin\theta \\ e_3e_1 - e_2\sin\theta & e_3e_2 + e_1\sin\theta & e_3^2 + \cos\theta \end{bmatrix}$$

2.6.4 Time derivatives

To relate orientations to forces that cause them, it is useful to find angular velocities and accelerations of the structure. In a time-domain analysis, the time derivatives will have to be approximated using the finite difference method, which requires a time history of orientations. If this time history is given in terms of joint rotation vectors $(\phi, \theta, \psi)^T$, then the first derivative taken from it will represent a velocity vector $(\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ in the "configuration space" of these rotation angles, not in physical space.

To recover an angular velocity vector ω in physical space from the joint rotation velocity in configuration space, a matrix $\mathbf{M}_{GJ}$ is derived. It represents a transformation (not necessarily a rotation) from the physical global coordinate frame to the frame of the joint rotations. The columns of $\mathbf{M}_{GJ}$ describe the basis vectors $\hat{\imath}, \hat{\jmath}, \hat{k}$ after they have undergone the transformation.

$$\mathbf{M}_{GJ} = \begin{bmatrix} \mathbf{R}_z\mathbf{R}_y\hat{\imath} & \mathbf{R}_y\hat{\jmath} & \hat{k} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi & 0 \\ \cos\theta\cos\psi & \cos\psi & 0 \\ -\sin\theta & 0 & 1 \end{bmatrix}$$

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi & 0 \\ \cos\theta\cos\psi & \cos\psi & 0 \\ -\sin\theta & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

At small angles $(\phi, \theta, \psi)^T \approx (0, 0, 0)^T$, it can be shown that the angular velocity is approximately equal to the joint velocity.

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -\sin\psi \cdot \dot{\theta} + \cos\psi\cos\theta \cdot \dot{\phi} \\ \sin\psi\cos\theta \cdot \dot{\phi} + \cos\psi \cdot \dot{\theta} \\ -\sin\theta \cdot \dot{\phi} + \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} -0 \cdot \dot{\theta} + 1 \cdot \dot{\phi} \\ 0 \cdot \dot{\phi} + 1 \cdot \dot{\theta} \\ -0 \cdot \dot{\phi} + \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

Thus, the difference between a joint rotation velocity vector and the true angular velocity vector becomes significant only at large angular displacements. In this way, it can be considered a geometrical non-linearity, which in this thesis is accounted for.

The angular acceleration vector α can be found by differentiating the angular velocity ω, using the multiplication rule.

$$\alpha = \dot{\omega} = \mathbf{M}_{GJ} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \dot{\mathbf{M}}_{GJ} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2}$$

Here, the first term in the sum is the joint acceleration transformed from configuration space to physical space. The second term is the influence of the changing coordinate frame. Performing the differentiation of $\mathbf{M}_{GJ}$, expanding the multiplication, and collecting the terms, an expression is obtained for angular acceleration in terms of joint rotation, velocity and acceleration.

$$\alpha = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = \begin{bmatrix} -(\sin\psi\cos\theta \cdot \dot{\psi} + \sin\theta\cos\psi \cdot \dot{\theta})\dot{\phi} - \sin\psi \cdot \ddot{\theta} + \cos\psi\cos\theta \cdot \ddot{\phi} - \cos\psi \cdot \dot{\psi}\dot{\theta} \\ -(\sin\psi\sin\theta \cdot \dot{\theta} - \cos\psi\cos\theta \cdot \dot{\psi})\dot{\phi} + \sin\psi\cos\theta \cdot \ddot{\phi} - \sin\psi \cdot \dot{\psi}\dot{\theta} + \cos\psi \cdot \ddot{\theta} \\ -\sin\theta \cdot \ddot{\phi} - \cos\theta \cdot \dot{\phi}\dot{\theta} + \ddot{\psi} \end{bmatrix}$$

At small angles $(\phi, \theta, \psi)^T \approx (0, 0, 0)^T$, one should no longer neglect the difference between joint rotation acceleration and physical angular acceleration.

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} \approx \begin{bmatrix} -(0 \cdot \dot{\psi} + 0 \cdot \dot{\theta})\dot{\phi} - 0 \cdot \ddot{\theta} + 1 \cdot \ddot{\phi} - 1 \cdot \dot{\psi}\dot{\theta} \\ -(0 \cdot \dot{\theta} - 1 \cdot \dot{\psi})\dot{\phi} + 0 \cdot \ddot{\phi} - 0 \cdot \dot{\psi}\dot{\theta} + 1 \cdot \ddot{\theta} \\ -0 \cdot \ddot{\phi} - 1 \cdot \dot{\phi}\dot{\theta} + \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \ddot{\phi} - \dot{\theta}\dot{\psi} \\ \ddot{\theta} - \dot{\phi}\dot{\psi} \\ \ddot{\psi} - \dot{\phi}\dot{\theta} \end{bmatrix}$$

In order to disregard geometrical non-linearities when calculating angular accelerations, an additional assumption must be made. The joint rotation velocities must be considered small, so that $(\dot{\phi}, \dot{\theta}, \dot{\psi})^T \approx (0, 0, 0)^T$. Then, the changing coordinate frame influence in Equation 2 is negligible and the approximation is valid:

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} \approx \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix}$$

3 Method

In this chapter, the two software programs are briefly introduced. A description is given of the model, as it is defined inside SIMA before any coupling is attempted. Then, a derivation is given for the coupling force necessary to connect the two simulations. The programmatic details of applying this coupling force are described. Necessary prerequisites for the coupling to work, such as initial conditions, are discussed. Different functionalities available in SIMA and SAMS are explored and a choice is made about their use. After making sure the prerequisites are in place, the SIMA model is modified to allow coupling. An ice field is defined in SAMS, along with structure properties. Finally, a simulation matrix is defined with varying environmental parameters.

3.1 SIMA

SIMA is a software package developed by DNV (DNV, 2021c) composed of several numerical computational modules. Two of them, in particular, were used in the simulations performed in this thesis: SIMO and RIFLEX. Their integration is described in detail in Bachynski, 2014. It is illustrated in Figure 8.

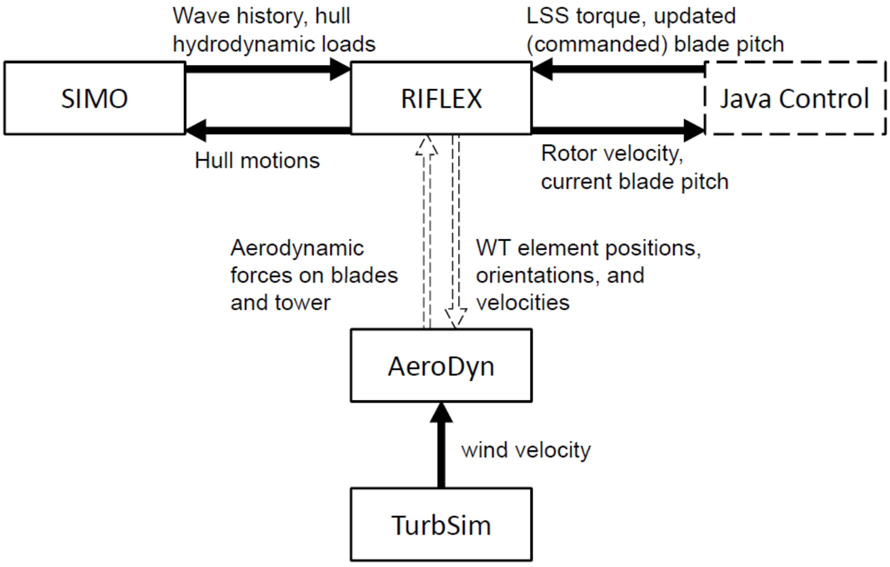

Wave history, hull hydrodynamic loads
LSS torque, updated (commanded) blade pitch
SIMO
RIFLEX
Java Control
Hull motions
Rotor velocity, current blade pitch
Aerodynamic forces on blades and tower
WT element positions, orientations, and velocities
AeroDyn
wind velocity
TurbSim

Figure 8: SIMO-RIFLEX-AeroDyn overview.

Source: Bachynski, 2014

In the context of Figure 8, the contribution of this thesis will be to connect the SIMO module with another software, SAMS.

3.2 SAMS

The Simulator for Arctic Marine Structures (SAMS) is a software package developed by Arctic Integrated Solutions AS (ArcISo). Some of the founding principles for the software are laid out in Lubbad and Løset, 2011, where a focus was made on ship-ice interaction. After implementing the non-smooth discrete element method (NDEM) as described by van den Berg et al., 2018, the software underwent a number of additional expansions and validation studies. The building blocks of SAMS are illustrated in Lubbad et al., 2018.

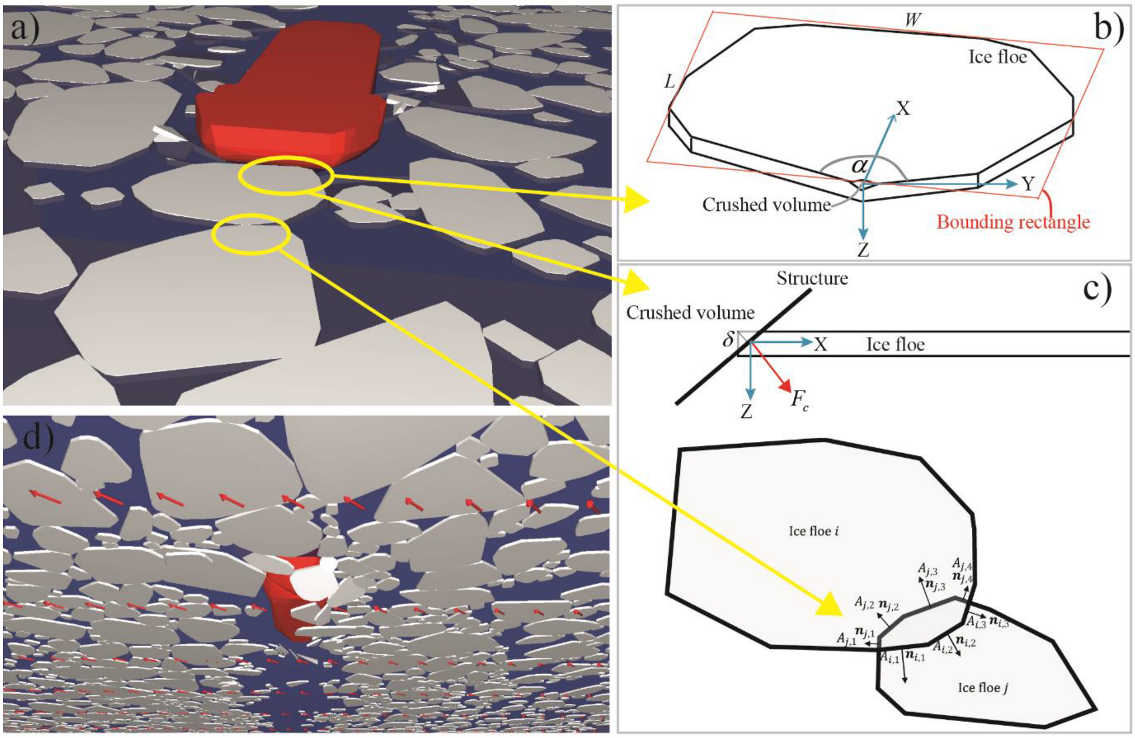


Figure 9: Illustration of different modules within SAMS: a) the simulation environment; b) the fracture module; c) the NDEM or multi-body dynamic module illustrating the non-rigid contact due to ice crushing at the contact interface; d) illustration of the fluid domain, e.g. current flow, with velocity vectors.

Source: Lubbad et al., 2018

Here, the fracture module (b) implements the different failure modes described in Section 2.2.1.

3.3 Model description

The floating wind turbine used in this thesis was developed in Wang, 2014. In the study, a semi-submersible platform was designed to support the DTU 10-MW reference wind turbine described in Bak et al., 2013. The model for coupling analysis was developed in SIMA, using three of its modules for different aspects of the analysis as shown in Figure 10.

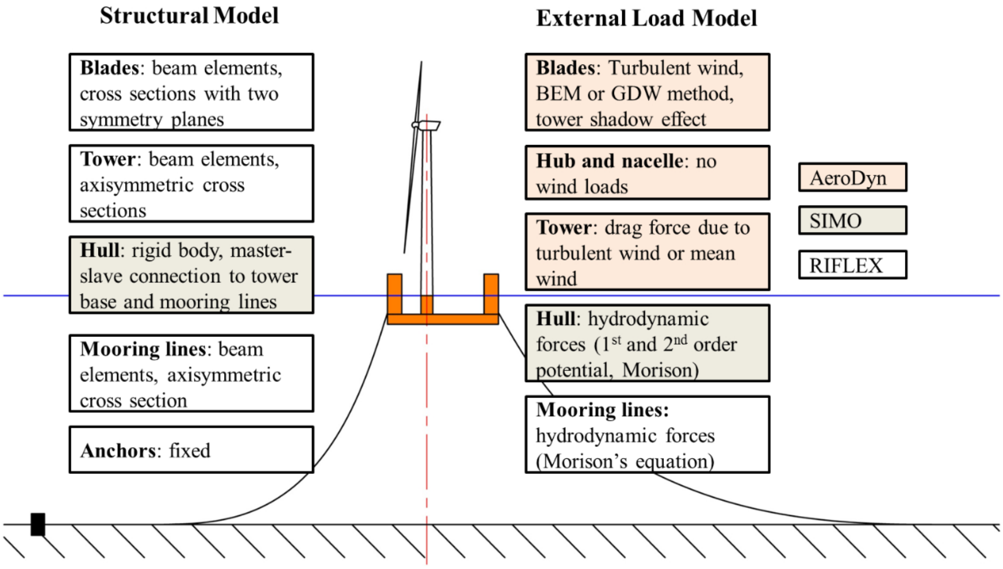

Structural Model
Blades: beam elements, cross sections with two symmetry planes
Tower: beam elements, axisymmetric cross sections
Hull: rigid body, master-slave connection to tower base and mooring lines
Mooring lines: beam elements, axisymmetric cross section
Anchors: fixed
External Load Model
Blades: Turbulent wind, BEM or GDW method, tower shadow effect
Hub and nacelle: no wind loads
Tower: drag force due to turbulent wind or mean wind
Hull: hydrodynamic forces (1st and 2nd order potential, Morison)
Mooring lines: hydrodynamic forces (Morison's equation)
AeroDyn
SIMO
RIFLEX

Figure 10: Description of the structural model and external load model of the semi-submersible floating wind turbine.

Source: Wang, 2014

On the left side of Figure 10, the focus of this this thesis is the hull rigid body. On the right side, the goal is to add a new force to the list, ice forces. Both are related to the SIMO module. From the perspective of the rigid body, the forces from the other elements of the model, such as mooring, are also external, just like ice.

3.3.1 Mass and hydrostatic stiffness of the rigid floater

In this model, gravity was not calculated based on the mass of the floater and buoyancy was not calculated through its displaced volume. Instead, both were pre-calculated and assigned as static forces. The dynamic behavior from the interaction of these two forces was instead modeled through hydrostatic stiffness calculated from Equations 3.34-3.37 of Wang, 2014. Its values, as they are input to SIMA, are shown in Table 2.

Hydrostatic stiffness matrix coefficient	Description	Value
$\mathbf{K}_{3,3}$ [N/m]	Heave restoring coefficient	$2.907 \cdot 10^6$
$\mathbf{K}_{4,4}$ [N·m]	Roll restoring coefficient	$2.411 \cdot 10^9$
$\mathbf{K}_{5,5}$ [N·m]	Pitch restoring coefficient	$2.411 \cdot 10^9$

Table 2: Hydrostatic restoring coefficients constituting the stiffness matrix **K** of the floater

The equilibrium position, towards which the hydrostatic restoring force acts, coincides with the origin point of the global coordinate system in SIMA.

Despite not being used for calculating gravity, the mass matrix **M** is still necessary for calculations involving inertia. The mass matrix coefficients, calculated based on Table 4.1 of Wang, 2014, are presented in Table 3 as they are input to SIMA.

Mass matrix coefficient	Symbol	Description	Value
$\mathbf{M}_{1,1} = \mathbf{M}_{2,2} = \mathbf{M}_{3,3}$	m [kg]	Mass of the floater	$1.2642 \cdot 10^7$
$\mathbf{M}_{4,4}$	I_{xx} [kg·m^2]	Mass moment of inertia around the body X-axis	$6.414 \cdot 10^9$
$\mathbf{M}_{5,5}$	I_{yy} [kg·m^2]	Mass moment of inertia around the body Y-axis	$6.414 \cdot 10^9$
$\mathbf{M}_{6,6}$	I_{zz} [kg·m^2]	Mass moment of inertia around the body Z-axis	$1.1817 \cdot 10^{10}$

Table 3: Mass coefficients constituting the mass matrix **M** of the floater

3.3.2 Ice-breaking cones

Ice forces on the structure are reduced if ice fails in bending rather than crushing, as described in Section 2.2. For this purpose, a conical structure was added to each column of the semi-submersible structure. An initial guess was made for the geometry choice of the cone, illustrated in Figure 11.

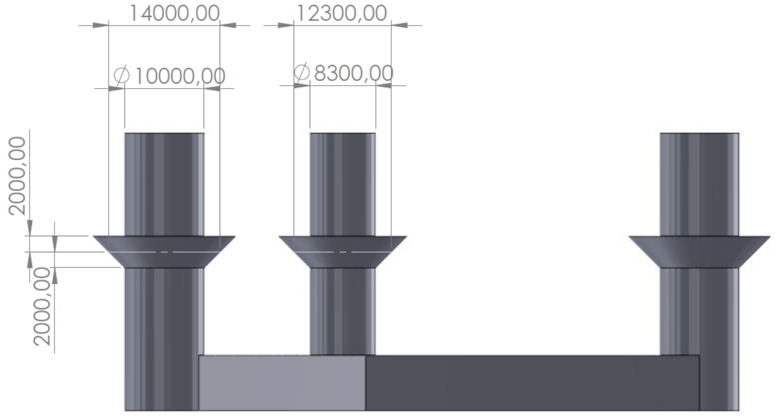


Figure 11: Dimensions of the ice-breaking cones [mm].

The sloped surface extends 2 m both below and above the design waterline. The validity of this choice is discussed in Section 4.3. This geometry was then used in SAMS for collision detection and buoyancy force calculation. For importing into SAMS, the geometry had to be represented as a mesh of triangular panels. Moreover, it was decomposed into convex parts. The mesh and division into parts is illustrated in Figure 12.

Despite the addition of the cones, neither the mass matrix nor the hydrostatic stiffness matrix were updated to reflect the increased mass or waterplane area. In further studies where the goal is accurate representation of a design scenario, both matrices should be updated to include these contributions.

3.3.3 Coordinate frames

Without coupling of SIMA and SAMS, there would be two coordinate frames to keep track of: the SIMA global coordinate frame, which is fixed relative to the environment, and the SIMA body local coordinate frame fixed to the rigid semi-submersible. With the addition of SAMS, one must also keep track of the SAMS global and SAMS body local coordinate frame. In the case of successful coupling, the SIMA global and local coordinate frames should have the same orientation and position, relative to each other, as in SIMA. The

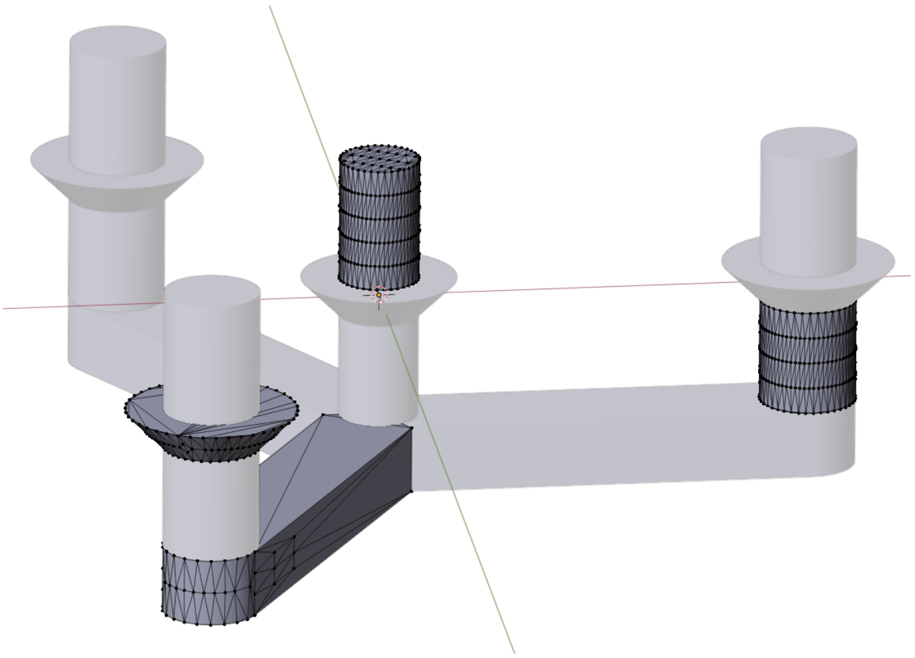


Figure 12: Meshing of the semi-submersible geometry.

SIMA and SAMS global coordinate frames, however, are never equal in this study. In SIMA, the global Z-axis is defined as pointing to the sky. In SAMS, the global Z-axis is defined as pointing to the ground. This 180-degree difference is illustrated in Figure 13.

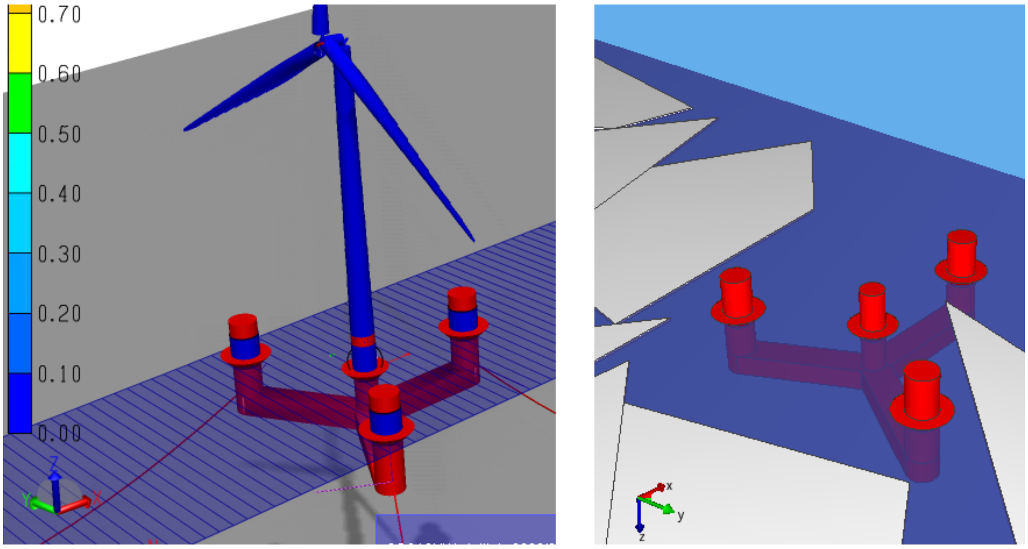


Figure 13: Difference between global coordinate frames in SIMA (left) and SAMS (right).

3.4 Equation of motion and coupling force

At a given timestep k with time t, the rigid body of the floater will have a kinematic state consisting of displacement and velocity vectors $(\mathbf{x}_k, \dot{\mathbf{x}}_k)$. Since the models in SAMS and SIMA are different, then without coupling the two software packages will find two conflicting kinematic states $(\mathbf{x}_{SIMA_k}, \dot{\mathbf{x}}_{SIMA_k})$ and $(\mathbf{x}_{SAMS_k}, \dot{\mathbf{x}}_{SAMS_k})$ for every timestep k.

SINTEF, 2018 gives the equation of motion solved by SIMA as

$$(\mathbf{m} + \mathbf{A}_\infty)\ddot{\mathbf{x}} + \mathbf{D}_1\dot{\mathbf{x}} + \mathbf{D}_2\mathbf{f}(\dot{\mathbf{x}}) + \mathbf{K}\mathbf{x} + \int_0^t \mathbf{h}(t-\tau)\dot{\mathbf{x}}(\tau)\mathrm{d}t = \mathbf{q}(t, \mathbf{x}, \dot{\mathbf{x}}) \tag{3}$$

Here, $\mathbf{m}$ is the body mass matrix, $\mathbf{A}_\infty$ is the infinite frequency added mass matrix, $\mathbf{D_1}$ is the linear damping matrix, $\mathbf{D_2}$ is the quadratic damping matrix, and $\mathbf{K}$ is the hydrostatic stiffness matrix. $\mathbf{h}(\tau)$ is the retardation function, dependent on the integration variable τ of the convolution integral. On the right side, $\mathbf{q}$ is the excitation force vector consisting of several components.

$$\mathbf{q}(t, \mathbf{x}, \dot{\mathbf{x}}) = \mathbf{q}_{WI} + \mathbf{q}_{WA}^{(1)} + \mathbf{q}_{WA}^{(2)} + \mathbf{q}_{CU} + \mathbf{q}_{add}$$

Here, $\mathbf{q}_{WI}$ is the wind drag force, $\mathbf{q}_{WA}^{(1)}$ is the first-order wave excitation force, $\mathbf{q}_{WA}^{(2)}$ is the second-order wave excitation force, $\mathbf{q}_{CU}$ is the current drag force. $\mathbf{q}_{add}$ contains any other forces such as wave drift damping, forces from slender elements such as the wind turbine tower and mooring fairleads, and ice forces. Most of these additional forces are also calculated by modules of SIMA. For example, the forces from slender elements are calculated by RIFLEX. For this reason, they will be referred to as $\mathbf{q}_{add_{SIMA}}$. The only exception is the ice force, which will be referred to separately as $\mathbf{q}_{ice}$.

$$\mathbf{q}_{add} = \mathbf{q}_{add_{SIMA}} + \mathbf{q}_{ice}$$

Equation 3 is solved by SIMA implicitly. The role of the coupling code developed in this thesis is to provide $\mathbf{q}_{ice}$ to SIMA with the help of SAMS.

$$\mathbf{q}_{ice} = f_{SAMS}(\mathbf{x}_{SIMA}, \dot{\mathbf{x}}_{SIMA}) \tag{4}$$

Here, f_{SAMS} represents the process of feeding in a kinematic state and getting in return an ice force vector. van den Berg et al., 2018 states that the ice contact forces are "only dependent on the penetration, and not the penetration velocity". This is a consequence of using the NDEM method. Thus, the velocity can be dropped from Equation 4, leaving only the displacement.

$$\mathbf{q}_{ice} = f_{SAMS}(\mathbf{x}_{SIMA})$$

To keep a consistent level of accuracy with the rest of SIMA calculations, $\mathbf{q}_{ice}$ should be calculated implicitly as well. At the time of writing this thesis, there are two software limitations preventing an implicit solution. First, the force exchange algorithm described in Section 3.6 supports only one force exchange per timestep. This excludes the possibility to find the ice forces iteratively. Second, even though SIMA performs its own iterations, it does not make the temporary solution $\mathbf{x}_{SIMA_i}$ available to the coupling code at the end of each iteration. For these reasons, the ice force must be found explicitly based on the kinematic state at the beginning of timestep k.

$$\mathbf{q}_{ice_k} = f_{SAMS}(\mathbf{x}_{SIMA_{k-1}}) \tag{5}$$

This is also a reason for a small delay in the ice force provided to SIMA. Given $\mathbf{x}_{SIMA_{k-1}}$ and having in memory $\mathbf{x}_{SIMA_{k-2}}$, SAMS will find the ice forces that were acting on the structure while it was moving from one kinematic state to the other. This, from the perspective of SIMA, should be the previous timestep. However, for lack of more recent information, the ice force will be used as if it were acting on the SIMA structure during the current timestep. Any possible inaccuracies resulting from this delay become less significant as the timestep duration becomes smaller. In this thesis, their influence is neglected.

In an analogous fashion, the process of feeding in an ice force to SIMA and getting the next kinematic state of the system may be written as a "black box" function f_{SIMA}.

$$(\mathbf{x}_{SIMA_k}, \dot{\mathbf{x}}_{SIMA_k}) = f_{SIMA}(\mathbf{q}_{ice_k}) \tag{6}$$

Alternating programmatic solving of Equation 5 and Equation 6 is the essence of the coupling algorithm described in Section 3.6.

The software capabilities of SAMS impose further constraints on Equation 5. The displacement $\mathbf{x}_{SAMS_k}$ of the structure cannot be given as an input argument. Instead, the kinematic state is found by solving a system of equations of motion described in Section 3.2. The software can only accept force inputs. Thus, a force vector $\mathbf{q}_{coupling}$ must be found such that

$$\mathbf{q}_{ice_k} = f_{SAMS}(\mathbf{x}_{SIMA_{k-1}}) = g_{SAMS}(\mathbf{q}_{coupling_k}) \tag{7}$$

Here, $g_{SAMS}(\mathbf{q}_{coupling_k})$ represents what is actually possible at the moment of writing this thesis: feeding in a coupling force vector and getting back an ice force vector. Equation 7 will be satisfied if the force vector $\mathbf{q}_{coupling}$, together with other forces acting on the SAMS structure, results in SAMS finding $\mathbf{x}_{SAMS_k} = \mathbf{x}_{SIMA_{k-1}}$. The solution is found implicitly in SAMS. Conversely, the coupling code is explicit. Thus, some simplifying assumptions should be made to enable estimation of $\mathbf{q}_{coupling}$.

To start, the linear and angular displacement at the end of the step is assumed to depend on the displacement at the beginning of the timestep $\mathbf{x}_{SAMS_{k-1}}$ and velocity $\dot{\mathbf{x}}_{SAMS_k}$ of the structure throughout the timestep k.

$$\mathbf{x}_{SAMS_k} = \mathbf{x}_{SAMS_{k-1}} + \dot{\mathbf{x}}_{SAMS_k} \cdot \Delta t \tag{8}$$

$\Delta t = t_k - t_{k-1}$ is the timestep duration. Velocity $\dot{\mathbf{x}}_{SAMS_k}$, in turn, is assumed to be determined by the previous velocity $\dot{\mathbf{x}}_{SAMS_{k-1}}$ and an impulse vector λ_k. The NDEM method used by SAMS implies that the structure receives the entire impulse λ_k instantly at the beginning of the timestep k.

$$\dot{\mathbf{x}}_{SAMS_k} = \dot{\mathbf{x}}_{SAMS_{k-1}} + \mathbf{M}_{SAMS}^{-1}\lambda_k \tag{9}$$

Here, $\mathbf{M}_{SAMS}$ is the mass matrix of the structure in SAMS. It is not equal to the total mass matrix $\mathbf{M} = (\mathbf{m} + \mathbf{A})$ used in the equation of motion that SIMA is solving, or the body mass matrix $\mathbf{m}$. The reason for that is elaborated in Section 3.8.

The total impulse λ_k received by the structure over a time Δt is expressed through the forces acting on the SAMS structure. The model in SAMS is set up such that the only forces acting on the structure are the ice forces $\mathbf{q}_{ice}$, coupling forces $\mathbf{q}_{coupling}$, and hydrostatic restoring forces $\mathbf{q}_{hydrostatic}$.

$$\lambda_k = \int_{t_{k-1}}^{t_k} (\mathbf{q}_{ice_k} + \mathbf{q}_{coupling_k} + \mathbf{q}_{hydrostatic_k})\mathrm{d}t \tag{10}$$

Recalling the instant momentum transfer assumption from Equation 9 and further assuming that any acting forces are constant throughout the timestep, Equation 10 can be rewritten as

$$\lambda_k = (\mathbf{q}_{ice_k} + \mathbf{q}_{coupling_k} + \mathbf{q}_{hydrostatic_k}) \cdot \Delta t \tag{11}$$

SAMS calculates $\mathbf{q}_{hydrostatic_k}$ according to Equation 4 of Lubbad and Løset, 2011. This force is not made available to the coupling code. Instead of repeating the process of finding the submerged volume of the structure and its center of buoyancy, $\mathbf{q}_{hydrostatic_k}$ is approximated as a hydrostatic stiffness force. This is possible because the stiffness matrix $\mathbf{K}$ declared in SIMA is known to the coupling code, as is the displacement $\mathbf{x}_{SAMS_{k-1}}$ at the beginning of timestep k.

$$\mathbf{q}_{hydrostatic_k} \approx \mathbf{K}(\mathbf{x}_{ref} - \mathbf{x}_{SAMS_{k-1}}) \tag{12}$$

Here, $\mathbf{x}_{ref}$ is the hydrostatic stiffness reference displacement. At this linear and angular position, the structure is assumed to be in hydrostatic equilibrium and the buoyancy forces balance the gravitational forces in all six degrees of freedom. In Section 4.2.5.1. of SINTEF, 2018, this stiffness reference position is referred to as $\mathbf{x}_0$. Here this notation is changed to avoid confusion with the initial displacement at the start of the simulation. In this thesis, the wind turbine model is placed so that its stiffness reference position coincides with the origin of both global coordinate systems in SIMA and SAMS. Thus, $\mathbf{x}_{ref} = (0,0,0,0,0,0)$ and Equation 12 simplifies to

$$\mathbf{q}_{hydrostatic_k} = -\mathbf{K}\mathbf{x}_{SAMS_{k-1}} \tag{13}$$

Explicit calculation means that $\mathbf{q}_{coupling_k}$ in Equation 11 needs to be given to SAMS before $\mathbf{q}_{ice_k}$ is known. If $\mathbf{q}_{ice_k}$ behaved as a smooth function, it would be possible to estimate it. For example, polynomial extrapolation could be used based on values at previous timesteps. However, as can be seen in Section 4, the ice forces are unpredictable enough to exclude any simple kind of estimation. For this reason, $\mathbf{q}_{ice_k}$ is assumed zero in this calculation. This means that $\mathbf{q}_{coupling_k}$ can only be considered reasonably accurate for timesteps where there are no ice forces. If the structure experiences ice forces for many consecutive timesteps, the accuracy of the coupling is expected to deteriorate. Conversely, if the structure experiences ice forces only in short bursts, the accuracy is expected to recover after every such burst. These hypotheses are further discussed in Section 4.1.

Inserting Equation 13 into Equation 11 and dropping $\mathbf{q}_{ice_k}$, the coupling force $\mathbf{q}_{coupling_k}$ can be expressed.

$$\mathbf{q}_{coupling_k} = \frac{\lambda_k}{\Delta t} + \mathbf{K}\mathbf{x}_{SAMS_{k-1}} \tag{14}$$

Equation 9 can be solved for λ_k and substituted into Equation 14.

$$\lambda_k = \mathbf{M}_{SAMS}(\dot{\mathbf{x}}_{SAMS_k} - \dot{\mathbf{x}}_{SAMS_{k-1}})$$

$$\mathbf{q}_{coupling_k} = \mathbf{M}_{SAMS}\left(\frac{\dot{\mathbf{x}}_{SAMS_k} - \dot{\mathbf{x}}_{SAMS_{k-1}}}{\Delta t}\right) + \mathbf{K}\mathbf{x}_{SAMS_{k-1}} \tag{15}$$

$\frac{\dot{\mathbf{x}}_{SAMS_k} - \dot{\mathbf{x}}_{SAMS_{k-1}}}{\Delta t}$ is the first-order backward finite difference of velocity. Recognizing that as an approximation for acceleration $\ddot{\mathbf{x}}_{SAMS_k}$, Equation 15 can be understood to represent a simplified discretised equation of motion for the SAMS structure. In this equation, the left-hand side represents the excitation force $\mathbf{q}_{coupling_k}$ and the right-hand side contains the inertial force $\mathbf{M}_{SAMS}\ddot{\mathbf{x}}_{SAMS_k}$ with the restoring force $\mathbf{K}\mathbf{x}_{SAMS_{k-1}}$. Thus, it is shown that the SAMS scenario has been set up with few enough forces that the structure can be treated as a simple undamped oscillator.

Equation 8 can be solved for $\dot{\mathbf{x}}_{SAMS_k}$ to express the velocity as a first-order backward finite difference of displacement. It can then be substituted into Equation 15.

$$\dot{\mathbf{x}}_{SAMS_k} = \frac{\mathbf{x}_{SAMS_k} - \mathbf{x}_{SAMS_{k-1}}}{\Delta t}$$

$$\mathbf{q}_{coupling_k} = \mathbf{M}_{SAMS} \left(\frac{\mathbf{x}_{SAMS_k} - \mathbf{x}_{SAMS_{k-1}}}{(\Delta t)^2} - \frac{\dot{\mathbf{x}}_{SAMS_{k-1}}}{\Delta t} \right) + \mathbf{K}\mathbf{x}_{SAMS_{k-1}}$$

Recalling that Equation 7 is satisfied when $\mathbf{x}_{SAMS_k} = \mathbf{x}_{SIMA_{k-1}}$, an expression is obtained for the coupling force vector.

$$\mathbf{q}_{coupling_k} = \mathbf{M}_{SAMS} \left(\frac{\mathbf{x}_{SIMA_{k-1}} - \mathbf{x}_{SAMS_{k-1}}}{(\Delta t)^2} - \frac{\dot{\mathbf{x}}_{SAMS_{k-1}}}{\Delta t} \right) + \mathbf{K}\mathbf{x}_{SAMS_{k-1}} \tag{16}$$

At the time of execution, all the terms on the right are known to the coupling code. Particularly, $\dot{\mathbf{x}}_{SAMS_{k-1}}$ can be either read from the SAMS output or calculated as a first-order finite backward difference of displacement.

$$\dot{\mathbf{x}}_{SAMS_k-1} = \frac{\mathbf{x}_{SAMS_k-1} - \mathbf{x}_{SAMS_{k-2}}}{\Delta t}$$

Substituting into Equation 16 and collecting terms, the coupling force becomes

$$\mathbf{q}_{coupling_k} = \mathbf{M}_{SAMS} \left(\frac{\mathbf{x}_{SIMA_{k-1}} - 2\mathbf{x}_{SAMS_{k-1}} + \mathbf{x}_{SAMS_{k-2}}}{(\Delta t)^2} \right) + \mathbf{K}\mathbf{x}_{SAMS_{k-1}} \tag{17}$$

Here, the fraction can be recognised as the second-order finite difference of displacement, an approximation for acceleration. Equation 16 and Equation 17 might be considered equivalent under the assumption that the velocity in SAMS is calculated close to the first-order finite difference of displacement. In practice, this assumption is not true. It is the displacement that is calculated from velocity in SAMS, not the other way around. Equation 16 works better for finding the coupling torque vector and ensuring coupling in the rotational degrees of freedom. Equation 17 works better for finding the coupling force vector and ensuring coupling in the translational degrees of freedom.

3.5 Coupling code

3.6 Coupling algorithm

To solve the equation of motion, both SIMA and SAMS are utilized to perform a coupled simulation.

Marintek, 2009 describes a possibility to extend the functionality of SIMO. This can be done through dynamically linked libraries (DLL), compiled from either FORTRAN or C. The latter programming language was chosen for its better support of TCP communications libraries (White et al., 2021).

A function is therefore written that SIMA calls on every timestep through the DLL interface. The descriptive header of the coupling code is included in Section 6.1. The code is compiled with the Intel C/C++ Compiler (Intel, 2021). The DLL interface is described in more detail in Appendix D.

The goal of the function is to calculate the coupling force $\mathbf{q}_{coupling}$, send it over TCP to SAMS, retrieve the resulting ice forces, and save them to the internal memory of SIMA. After completing its run, the coupling code writes the obtained information to a .csv file, using an open-access library (Tal H, 2021).

3.7 TCP communication between SAMS and SIMA

The TCP protocol, as well as its implementation for SAMS, is thoroughly described by its authors in Rolandsen and Hoel, 2018. The overall scheme is shown in Figure 14.

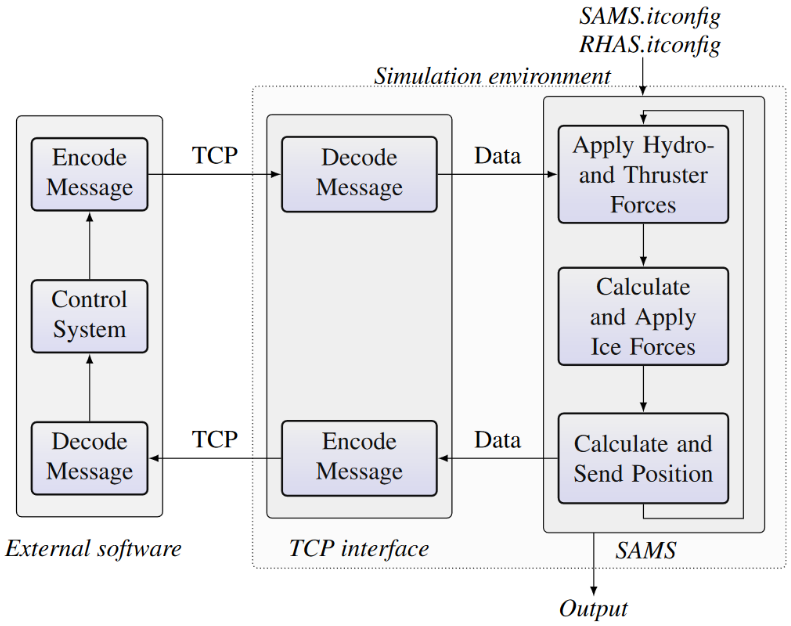


Figure 14: High level overview of the SAMS simulation environment together with the external software SIMA.

Source: Rolandsen and Hoel, 2018

Originally, the TCP interface was written with a dynamically controlled ship in mind. In the scope of this thesis, SIMA and its simulation environment, together with the coupling code, take the role of the "Control System" on Figure 14. In place of "Hydrostatic and Thruster Forces", the coupling force $\mathbf{q}_{coupling}$ as calculated in Section 3.4 is sent to SAMS. There was some inconsistency in handling the last timestep, described in Appendix A.

Rolandsen and Hoel, 2018 mention two possible methods of encoding and decoding the message. In this implementation, the bytestream option was chosen due to its simplicity. As a consequence, it is not possible to simulate more than one structure at a time to have contact with the ice. Additionally, only central forces and torques are used. Even though SAMS accepts an arbitrary number of point forces, this functionality is left unused in this thesis.

The role of SAMS in this coupling is to calculate the ice forces on the structure for a given position and orientation of the structure in an ice field. The simplest procedure therefore would be to feed in positions and get out forces. SAMS does not currently accept forced positions unless they are part of a predetermined track, known entirely before the start of the simulation.

Another challenge in finding position and orientation-dependent ice forces from SAMS is the fact that despite calculating them, SAMS does not actually communicate them over TCP. Instead, it writes them to a .txt result file some time after finishing the calculating of each timestep. The process of obtaining these ice forces from the file is described in Appendix B.

3.8 Initial conditions

The initial conditions in SIMA are set through the GUI.

The initial conditions in SAMS are also set through the GUI. There is one key difference, however. There is a field for inputting the initial vertical coordinate, but it is in fact ignored by the program. Instead, a vertical position is found at which the structure is in hydrostatic equilibrium. The simulation is then started with the structure at that vertical displacement, regardless of user input.

If the initial condition forced by SAMS happens to be different from the initial condition set in SIMA, the coupling will not work as intended. The initial conditions are almost guaranteed to happen, since the models differ considerably between programs. However, since the goal in SAMS is anyway to force a displacement, the mass of the structure does not have to correspond to the real mass of the structure. Instead, in this thesis the mass in SAMS was declared such that the structure would match initial positions with SIMA.

In SIMA, the vertical equilibrium position of the floater is determined by the buoyancy force described in Section 2.4.4 and gravitational force. The contributions to the gravitational force come from elements that have a declared mass in the SIMA model. They are summarised in Table 4.

Rigid body	Linetype	Location	Total mass [kg]
	shaft_lt	Rotor shaft	7
	bl_ecc	Rotor shaft	2
	nac_lt	Nacelle	3
	bl_foil	Blades	41721
	tower_lt	Tower	628440
	moor	Mooring lines	409200
SEMI		Floater	12642000
Total			**13721374**

Table 4: Elements with mass in the SIMA model of the floating wind turbine

It is assumed that at the start of the simulation, both structures have not only zero displacement from origo, but also zero velocity and acceleration.

3.9 Intermittent output of timestep results

The time-domain analysis is performed in timesteps of duration Δt. To simplify the analysis, the timestep duration is kept the same both in SIMA and SAMS. However, SIMA does not refer to this smallest time increment as a step. Instead, the discrete time increment is called a substep. A step is a larger unit of time, consisting of a fixed number of substeps. It is used to avoid having to write the results of every time increment, especially in cases of long simulations and short timestep length, resulting in a large number of substeps. Instead, while every substep is simulated, only information from every step is output.

Something similar is available in SAMS. The timestep duration is set through the frequency. This frequency determines the smallest time increment which is used in the simulation. To avoid printing the results of every timestep, the output can be written only after every n substeps. This is achieved by setting the ”output time step stride” option in SAMS.

Alternatively, one can also specify a TCP update frequency that is different from the simulation frequency. For the code in this thesis, this additional functionality is not used and every timestep is written to file both in SIMA and in SAMS. This means that in SIMA, every ”step” contains exactly one ”timestep”. In SAMS, the ”time step stride” is set to 1. It could be possible to add support for intermittent output, but in this version of the code this support has not been added.

3.10 Iterations

Both SIMA and SAMS can make several attempts to solve the equations of motions for one timestep. These iterations are repeated until the solution satisfies the equations of motions to an acceptable degree. It could be possible to run the coupling code on every iteration of SIMA.

However, it is not possible to have a TCP exchange on every iteration with SAMS. For this reason, the coupling code is only run on the very first iteration of every substep. For subsequent iterations of the same substep, the ice forces are used that were obtained from the TCP exchange of the first iteration.

3.11 Simulation setup

3.11.1 Modifications of the CSC 10 MW model

In order to enable coupled analysis of the CSC 10 MW model in SIMA, some modifications were made to it.

Storage of body forces was enabled in the dynamic calculation properties.

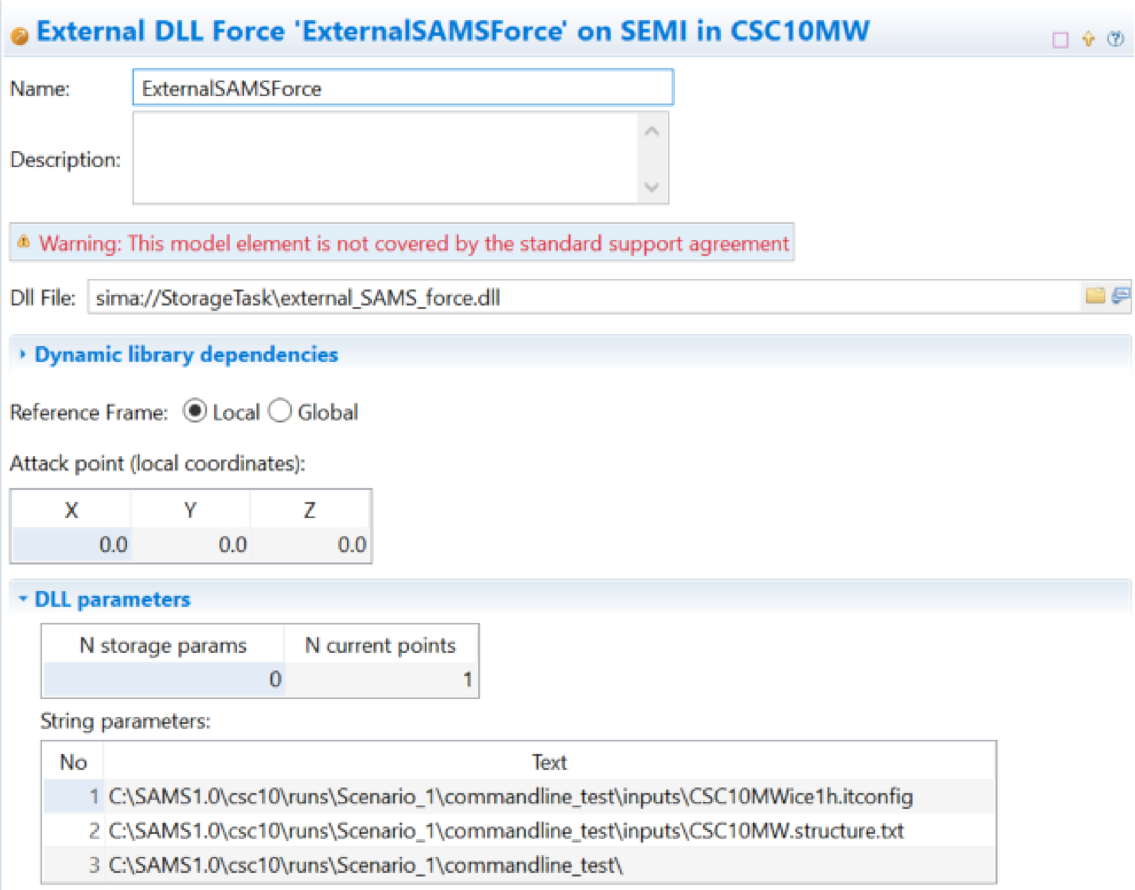
An ”External DLL Force” was declared in the model description, pointed to the location of the compiled coupling code. Additional provided locations included the SAMS result folder, the .itconfig file, and the .structure.txt file. For future use reference, an example of the relevant settings is provided in Figure 15.

It was then checked that the body position in SIMA, the hydrostatic stiffness reference, and the position in structure.txt are all the same position. To enable assumptions that lead to Equation 13, this position was chosen as $(0, 0, 0)$. Orientation was treated analogously.

To enable a simulation study, common variables were set up to control the simulation duration and timestep.

According to ArcISO, 2020, the SAMS simulation must be started before the SIMA simulation.

A structure.txt file was placed in the SAMS results folder. Its contents are more closely described in Section 3.11.3. It was checked that it contains an accurate reference to the .obj file containing the floater geometry described in Section 3.3.2. The exact same .obj file could not be reused, however. Instead, a file had to be used where the same geometry is rotated 180 degrees around the global X axis.



External DLL Force 'ExternalSAMSForce' on SEMI in CSC10MW

Name: ExternalSAMSForce

Description:

Warning: This model element is not covered by the standard support agreement

Dll File: sima://StorageTask\external_SAMS_force.dll

Dynamic library dependencies

Reference Frame: ● Local ○ Global

Attack point (local coordinates):

X	Y	Z
0.0	0.0	0.0

DLL parameters

N storage params	N current points
0	1

String parameters:

No	Text
1	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\inputs\CSC10MWice1h.itconfig
2	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\inputs\CSC10MW.structure.txt
3	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\

Figure 15: Configuration of the external DLL force in SIMA.

It was verified that the structure.txt contains an accurate structural mass m_{SAMS} according to Section 3.8. Since the radii of gyration do not influence the vertical equilibrium position in SAMS, their choice is arbitrary. The coupling code will use whatever radii of gyration to calculate an appropriate coupling force. To not deviate too much from the SIMA model, the radii of gyration $\mathbf{r}_{SAMS}$ were calculated from the SIMA body mass, that is

$$\mathbf{r}_{SAMS} = \sqrt{\frac{\mathbf{I}_{SIMA}}{m_{SAMS}}}$$

Here, $\mathbf{I}_{SIMA}$ are mass moments of inertia, coefficients in the SIMA body mass matrix $\mathbf{m}$.

The .itconfig file was placed in the SAMS inputs folder. Its contents are more closely described in Section 3.11.3. The .itconfig file contains a reference to the structure.txt file.

A .bat file was created in the SAMS results folder for ease of execution. Inside, it was pointed to the .itconfig file. When run, user input is still needed. For example, the program asks for confirmation if there are ice floes intersecting the floater in the very first step. There warning can be turned off in SAMS, but were not for this thesis.

The algorithm to find the newest result files in the SAMS result folder is described in Appendix B. However, it is not robust enough for complicated naming schemes. For this reason, it is advisable to remove any previous result files from the SAMS result folder before starting the simulation.

3.11.2 Ice field properties

In a typical SAMS simulation, the ice field covers the positive range of the X-axis. If the structure is simulated to move with a constant speed through the ice field, then it has a positive velocity in X; if the structure is simulated be stationary or moored, then some forces have to be applied to the ice field that would make it move in the negative X direction.

This is contrary to the way SIMA is set up. In SIMA, the wind blows in the positive X direction and the wind turbine is oriented accordingly. Thus, it was decided to generate an ice field that moves with the wind. The ice floes start from the negative side of the X domain and move in the positive X direction, towards the wind turbine and past it. In Figure 16, the X-axis is directed towards the right.

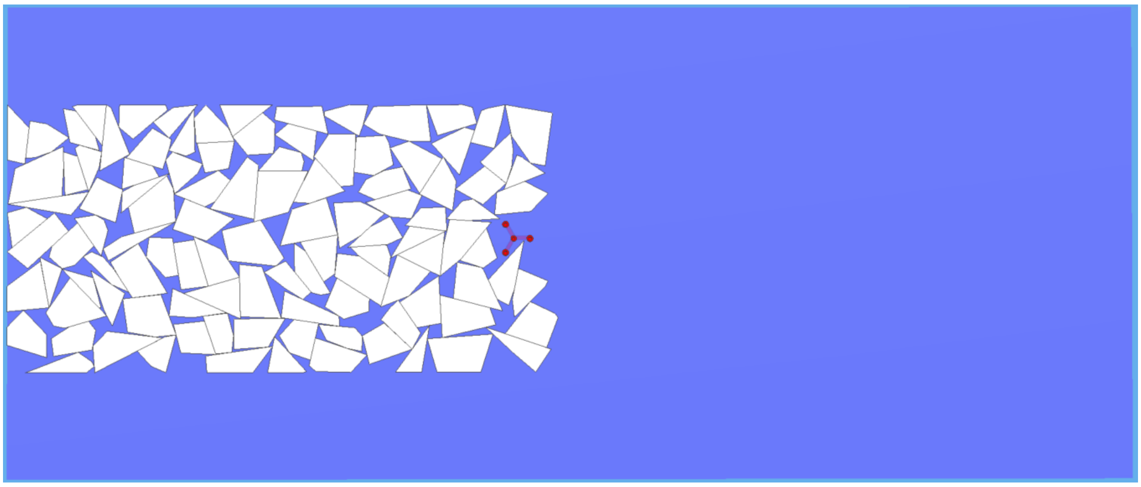


Figure 16: Floater in front of ice field.

The ice floes were generated so that their size distribution would follow a power law.

$$P(D) = \alpha D^{-\beta}$$

In this power law, D is the mean caliper diameter (MCD) of an ice floe and $P(D)$ is its probability of occurrence in the ice field. α is the ice concentration. The ice concentration is 0 for completely open water and 1 for complete ice cover of the surface. β is an arbitrary distribution parameter. For the ice field in this study, the ice field parameters and ice mechanical properties were assumed according to Table 5.

The choice of these parameters was informed by previous work such as Pettineo, 2021 and Afanasjev, 2021. Choosing the ice-air skin friction coefficient as zero means that in this thesis, the ice field was only driven by the current.

3.11.3 Global parameters of the simulation in SAMS

The SAMS simulation parameters are stored in two files. One governs the simulation and environment, while the other describes the structure and TCP interface.

In the structure.tcp file, the mass properties are not given as a mass matrix. Instead, they were given as one mass and radii of gyration. This carries the implied assumption that the mass is concentrated in the volumetric center of the body.

Parameter	Value
Field dimensions, $X \cdot Y$ [m· m]	2800·740
Ice field concentration [1]	0.8
Power law parameter β [1]	2.5
Minimum MCD [m]	100
Maximum MCD [m]	200
Ice density [kg/m^3]	910
Crushing specific energy [Pa]	$2.0 \cdot 10^6$
Young's modulus [Pa]	$5.0 \cdot 10^9$
Poissons's ratio [1]	0.3
Fracture toughness [Pa·m$^{0.5}$]	$1.5 \cdot 10^5$
Flexural strength [Pa]	$5.0 \cdot 10^5$
Tensile strength [Pa]	$5.0 \cdot 10^5$
Ice-ice friction coefficient [1]	0.15
Ice-structure friction coefficient [1]	0.15
Ice-water skin friction coefficient [1]	$5.0 \cdot 10^{-3}$
Ice-water form drag coefficient [1]	(0.1, 0.1, 0.5)
Ice-air skin friction coefficient [1]	0

Table 5: Statistical properties of the ice field and mechanical properties of ice

For ice-structure interactions to occur, there needs to be relative movement between the structure and the ice. There are two ways of achieving this. One way is to simulate the structure as advancing into the ice. This is the relatively simpler way, and it can be realised by placing the ice field in a stationary position and assigning a predefined velocity to the structure. In this case, the structure will have no kinematic response to the collision with ice.

Another way is to assign a stationary position to the structure and instead have the ice field advance. In this case, none of the positions are prescribed to wither the ice floes or the structure. Instead, both move due to forces applied to them. The ice floes move due to current, wind and structure reaction forces. The structure moves due to ice forces, possibly linear mooring forces, and any external forces applied through TCP. This latter option is used in this thesis.

To maintain setup consistency, any wind and current that cause ice movement in SAMS are also applied in the SIMA environment.

3.11.4 Environmental properties and simulation frequency

For the seawater density, a typical value of 1025 kg/m^3 was chosen. For air density, a typical value of 1.226 kg/m^3 was chosen. To maintain consistency, these values were used both in SIMA and in SAMS.

Other environmental values were varied in different combinations, as shown in Table 6. The simulation frequency f was varied for the most complicated cases between 5, 10 and 20 Hz. For all other cases, a default of 10 Hz was used.

Scenario	Ice in SIMA	H_s [m]	U_0 [m/s]	Wind model	f [Hz]
01	No	0.01	0.0	-	10
02	No	0.01	11.4	Constant	10
03	No	0.01	11.4	Turbulent	10
04	No	0.30	0.0	-	10
05	No	0.30	11.4	Constant	10
06	No	0.30	11.4	Turbulent	10
07	Yes	0.01	0.0	-	10
08	Yes	0.01	11.4	Constant	10
09	Yes	0.01	11.4	Turbulent	10
10	Yes	0.30	0.0	-	10
11	Yes	0.30	11.4	Constant	5
12	Yes	0.30	11.4	Constant	10
13	Yes	0.30	11.4	Constant	20
14	Yes	0.30	11.4	Turbulent	5
15	Yes	0.30	11.4	Turbulent	10
16	Yes	0.30	11.4	Turbulent	20
17	Yes	0.30	11.4	Turbulent	25

Table 6: Simulation conditions for different simulated scenarios

The simulations were run in both coupled and uncoupled regime. For the uncoupled simulations, both programs started from the same initial conditions but simulated completely different processes. The simulation in SIMA represented a wind turbine in ice-free sea, while the SAMS simulation represented an unmoored, free-floating semi-submersible in pack ice.

The wind conditions were varied between constant uniform and turbulent wind, as well as no wind at all. For turbulent wind, the power exponent α was left as the default value 0.11 in SIMA. Veritas, 2010 recommends a value of 0.12. The first five minutes of the generated turbulent wind speed history are presented in Figure 17. This history was measured during the simulations at the nacelle.

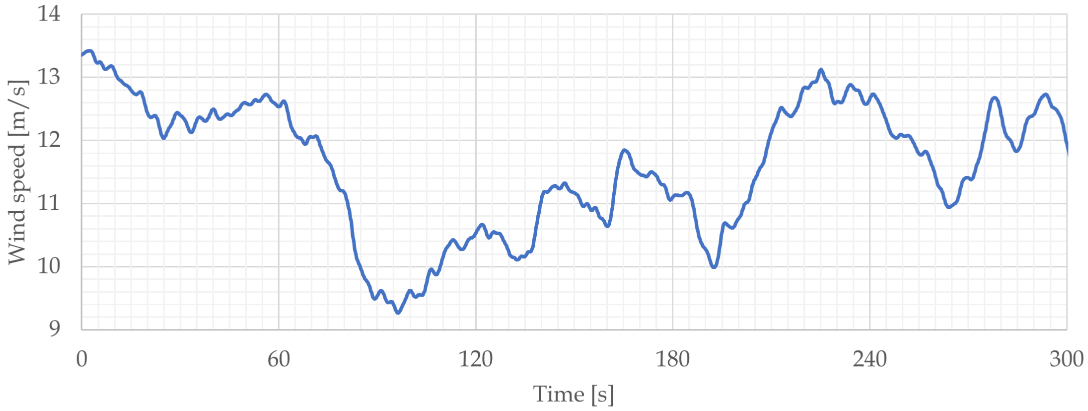


Figure 17: Turbulent wind speed in the global X direction.

It was assumed that the irregular seawaves follow the JONSWAP spectrum as described in SINTEF, 2018. The wave period was assumed as 12 s. The significant wave height was varied between 0.01 m and 0.3 m. Since SIMA does not allow a zero significant wave height, 0.01 m was used to represent a condition without waves. For most of the scenarios with waves, the same generated elevation history was used. Its first five minutes are presented in Figure 18.

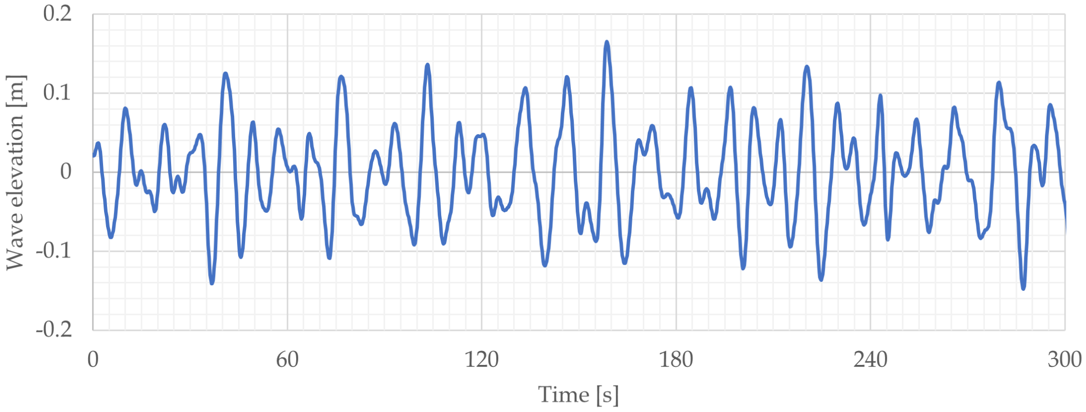


Figure 18: Wave elevation history generated in SIMA.

4 Results

In this chapter, it is judged whether the coupling algorithm was effective in ensuring similar kinematic response of the structure in both SAMS and SIMA. Special attention is paid to periods with long uninterrupted ice contact. Then, a sensitivity study is performed with regards to simulation frequency. Finally, the simulations are assessed by comparing coupled and uncoupled scenarios with different environmental setups.

4.1 Effectiveness of coupling

As an example, scenario 16 from Table 6 is chosen to illustrate the effect of the coupling on the floater displacements found in both SAMS and SIMA. In this scenario, there are the most forces acting on the floater: ice, waves, turbulent wind, as well as the flexible structure and current that is present in all other scenarios.

Comparisons of translational displacements between SIMA and SAMS are shown Figure 19 through Figure 21.

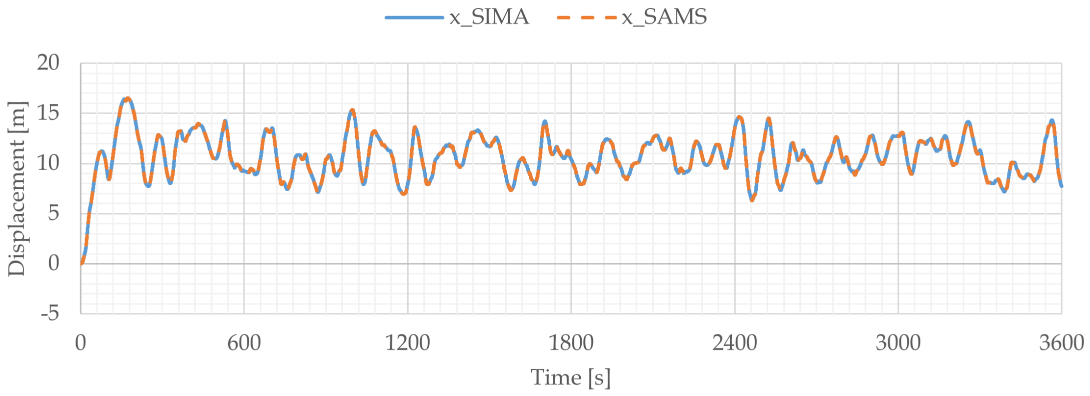


Figure 19: Coupling agreement in the SIMA global X direction.

It takes some minutes for the system to achieve a steady oscillatory state in surge, since the wind turbine rotor speed is zero at the beginning of the simulation. The displacements are simulated exactly the same even during this start-up period.

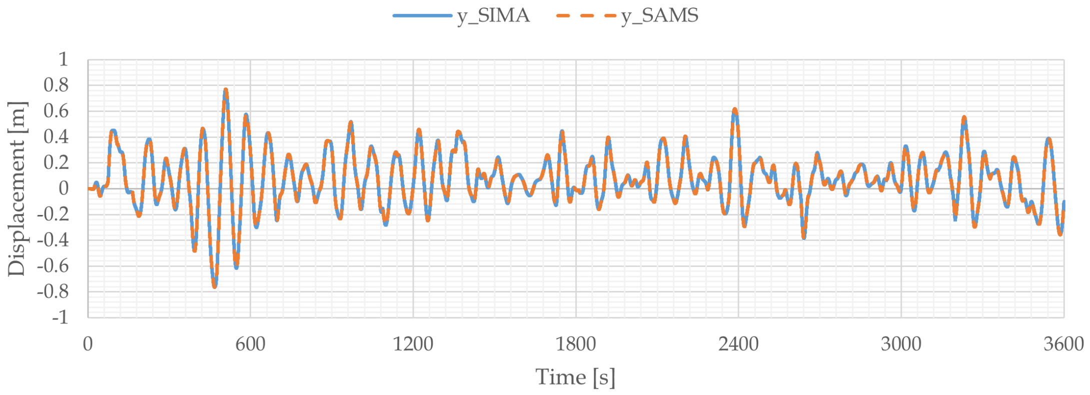


Figure 20: Coupling agreement in the SIMA global Y direction.

The heave response was found to have a higher frequency than surge or sway. Instead of the full hour, only 10 minutes toward the end of the simulation are presented in Figure 21. This makes it easier to see the lack of differences between the displacements that the two programs found.

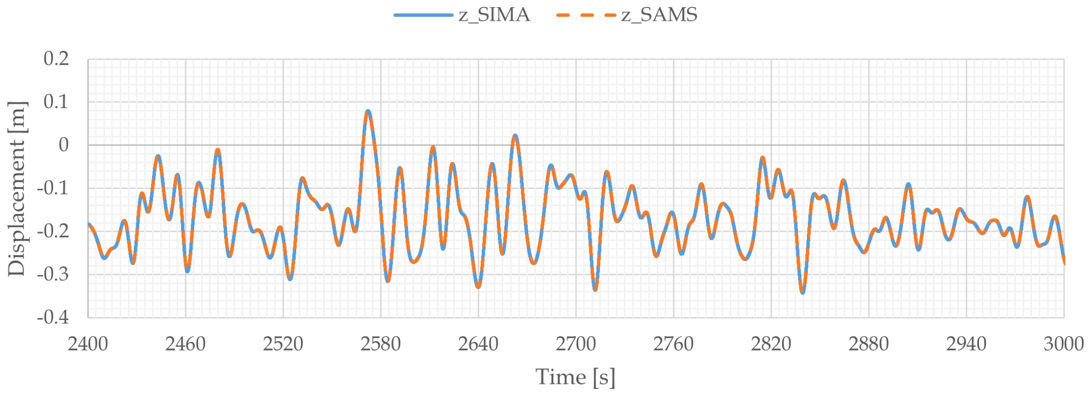


Figure 21: Coupling agreement in the SIMA global Z direction.

Comparisons of displacements in the two programs are shown in Figure 22 through Figure 24. Here, the orientation of the structure in SAMS was transformed from the SAMS axis-angle representation to the SIMA joint rotation representation, using the procedure described in Section 2.6.3. Thus, a comparison of orientations is possible. The joint rotations do not represent roll, pitch and yaw exactly, but can be assumed to approximate them at small angles. 10-minute periods are shown, just as for heave.

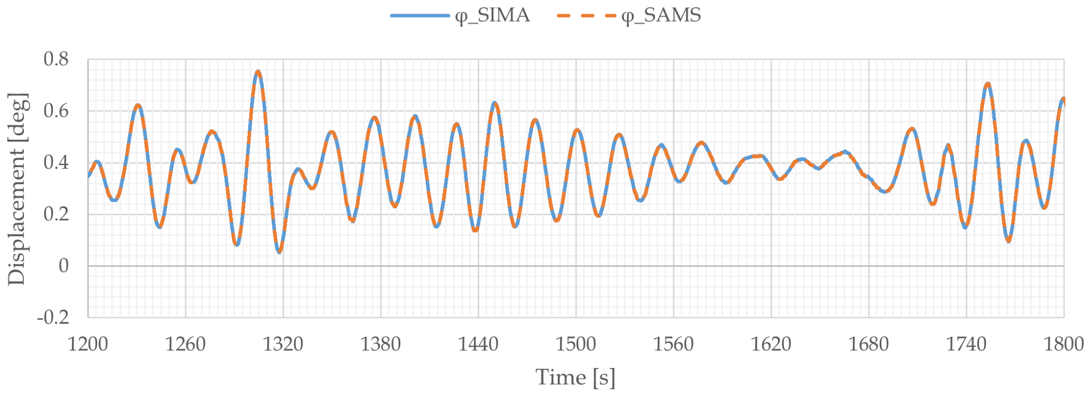


Figure 22: Coupling agreement in the SIMA joint ϕ direction.

There is a non-zero mean roll displacement, likely coming from asymmetry of the wind turbine around its local Y-Z plane.

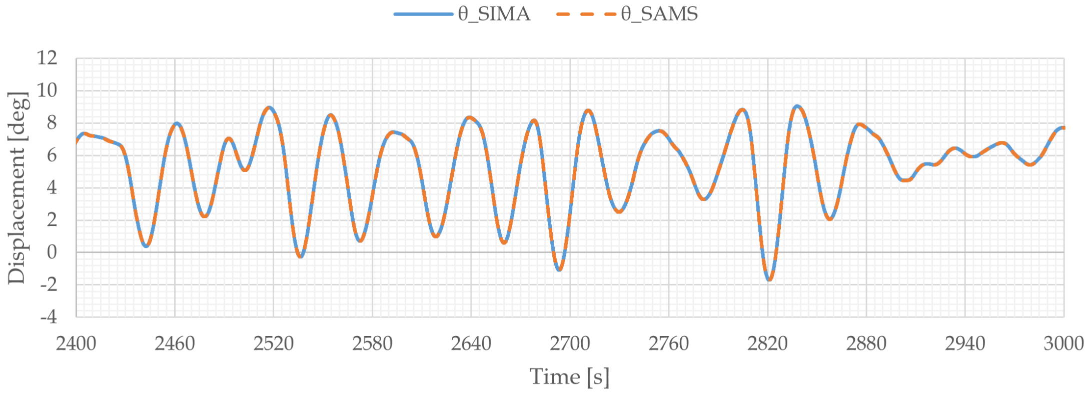


Figure 23: Coupling agreement in the SIMA joint θ direction.

Similarly, a nonzero mean pitch response is entirely expected. The wind, although turbulent, still exerts a mean thrust force on the wind turbine, leading to a pitch response that at times reaches 9 degrees. When that happens, the vertical displacement of the floater columns is significant. The windward columns of the floater will rise out of the water, while the leeward columns will be submerged. This effect has important consequences when designing the floater for ice loads, as shown later in Section 4.3.

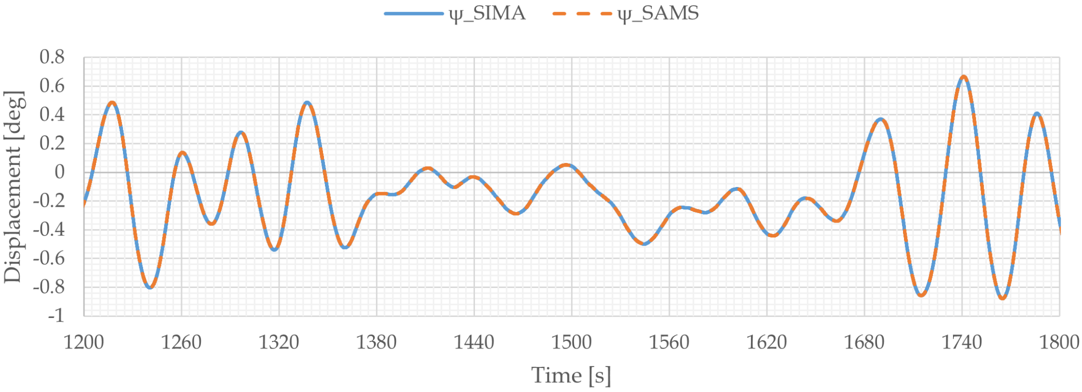


Figure 24: Coupling agreement in the SIMA joint ψ direction.

In yaw, the response is centered around a zero mean. It can be observed that in all six degrees of freedom, the displacements coincide completely. This indicates that the coupling code is successfully repositioning the structure in SAMS to follow displacements calculated in SIMA.

4.1.1 Effectiveness of coupling during uninterrupted ice contact

In the scenario 07 in Table 6, only a current was present without waves or wind. During the simulated hour, the longest uninterrupted ice contact period lasted close to 20 minutes, as shown in Figure 25.

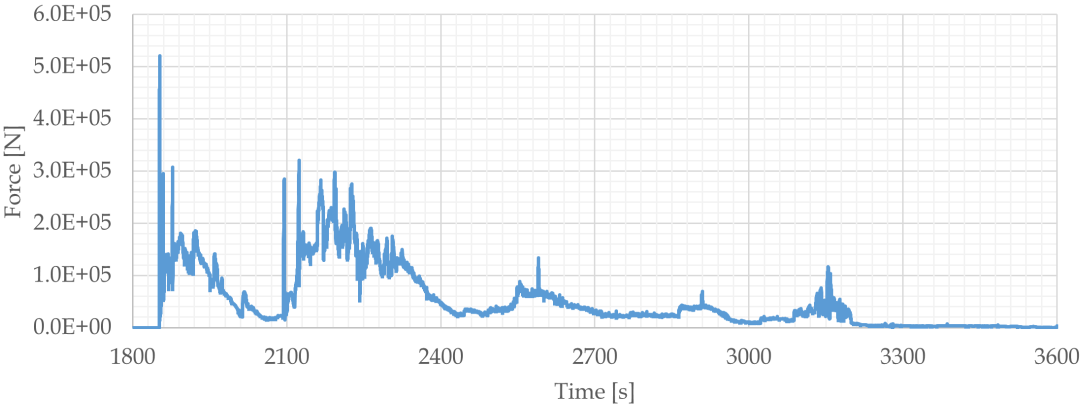


Figure 25: Ice force on the floater in the SIMA global X direction.

The influence of ice forces is well illustrated in Figure 26, where the displacement in X follows closely the ice force bursts.

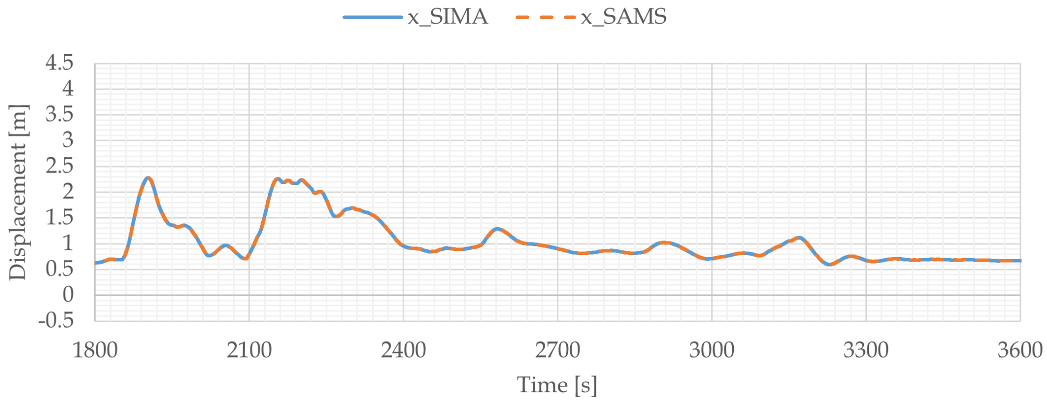


Figure 26: Coupling agreement in the SIMA global X direction.

As described in Section 3.4, the coupling algorithm disregards any potential ice forces when calculating the coupling force. Despite that, Figure 26 shows that the coupling quality is not deteriorated in any significant way. It may still be true that the zero ice force assumption introduces a small difference in kinematic states on every timestep. However, this difference will be implicitly taken into account when calculating the coupling forces for the next timestep. After all, the coupling force is obtained based on a target position $\mathbf{x}_{SIMA_{i-1}}$ and even if slightly incorrect, still precisely known starting position $\mathbf{x}_{SAMS_{i-1}}$. Thus, it can be said that the zero ice force assumption is sufficient for the purpose of this thesis and does not degrade coupling accuracy significantly.

4.2 Influence of simulation frequency

In Table 6, scenarios 14-16 all shared exactly the same environmental conditions. The only difference was in simulation frequency f, the reciprocal of timestep duration Δt. Scenarios 11-13 also share the exact same environmental setup. In their setup, the wind is constant. For scenarios 14-16, the wind is turbulent. The frequency dependence findings are largely the same, so only scenarios 14-16 with turbulent wind will be presented in this section. The computation times are presented in Table 7. The scenarios of interest are emphasised in bold.

Scenario	f [Hz]	Time [hh:mm]
11	5	00:58
12	10	03:14
13	20	13:09
14	**5**	**00:56**
15	**10**	**03:23**
16	**20**	**12:47**
17	25	20:33

Table 7: Time spent simulating one hour of a coupled scenario with wind and waves

After running the simulations, it became apparent that the turbulent wind and wave elevation history generated for scenario 17 was different from the histories in scenarios 14-16. For this reason, only the computation time is presented for scenario 17. Otherwise, this case is left out of the sensitivity study.

As suspected, a higher simulation frequency caused an increase in the computation time. The computation cost is increasing much faster than linearly. To find out the cause, profiling studies should be carried out for the two programs separately and together.

Qualitatively, the difference in simulation results can be seen by visual comparison. Having started from Figure 16, the simulation reaches three different states illustrated in Figure 27 through Figure 29.

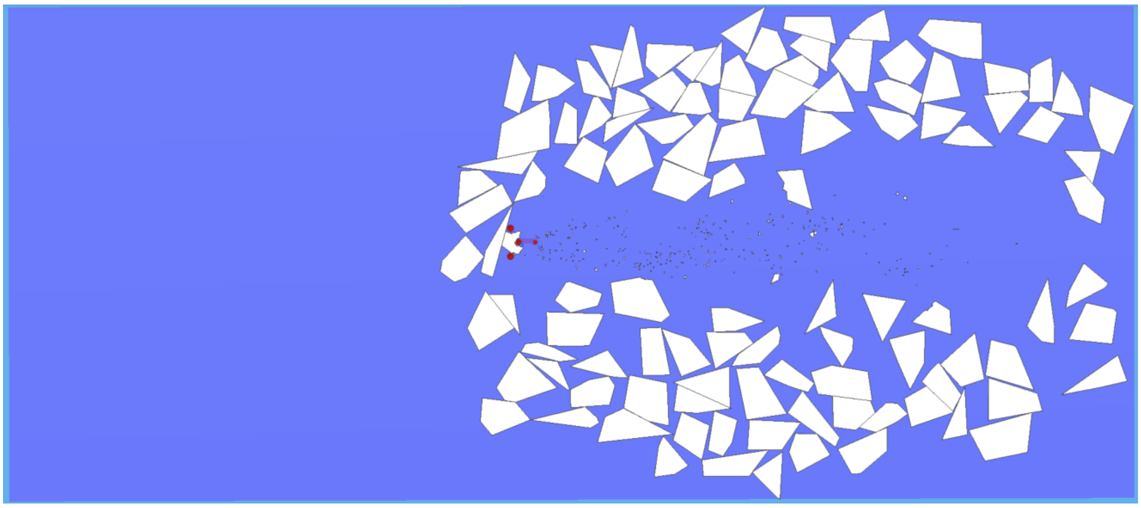


Figure 27: Final state of simulation with 5 Hz.

Most of the ice field has moved to the left, past the structure. A piece has broken off a floe (on Figure 16, the second one counting to the left of the floater). The piece has jammed between the front two legs of the structure. It has been carried all the way through the ice field in front of the floater. As the structure moves in pitch, its legs move up and down, breaking the jammed floe into smaller pieces every time.

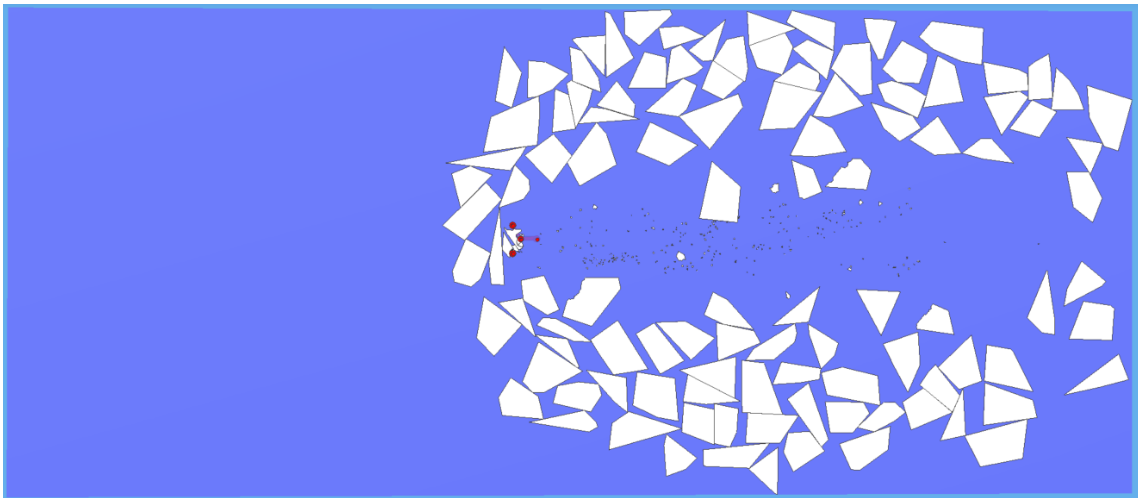


Figure 28: Final state of simulation with 10 Hz.

Visually, the dissimilarities increase to the left of the images. As the ice moves past the structure to the right, small differences in the simulation accumulate and magnify, leading to easily identifiable qualitative differences. Although in different positions and orientations, the individual ice floes can still be identified on both pictures for comparing.

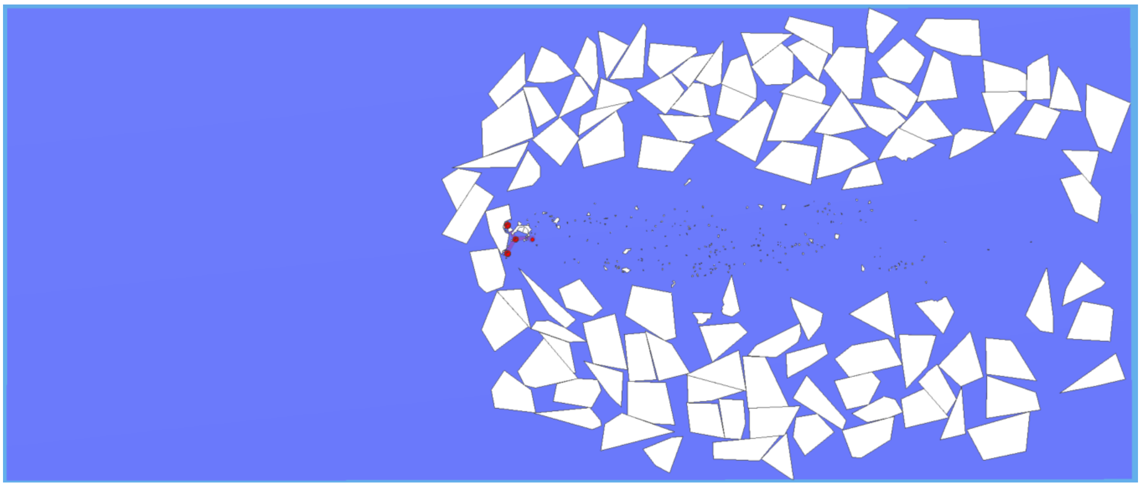


Figure 29: Final state of simulation with 20 Hz.

In all cases, the same ice floe jams in front of the structure. Interestingly, its degree of destruction is different between the simulations. In Figure 29, there is barely anything left of the jammed floe. Its pieces have cleared and moved past the structure. The floater has begun to collide with the next floe.

The differing level of destruction may be explained by the fact that with a higher frequency, the fracture detection code runs more times and thus fractures are more likely to occur often.

4.2.1 Surge

To gain a better understanding of the influence that the choice of frequency has on the simulation, the ice forces and corresponding motion histories are presented for each degree of freedom. The translational forces and displacements are presented in Figure 30 through Figure 35. All of these values are in the SIMA global coordinate frame. In the statistical summary that follows every time history graph, the five-minute time period 00:00-05:00 is cut away and not considered. This is done to avoid capturing transient effects that come from the simulation start-up. For example, the wind turbine rotor takes some time to speed up from a halt in the beginning of the simulation.

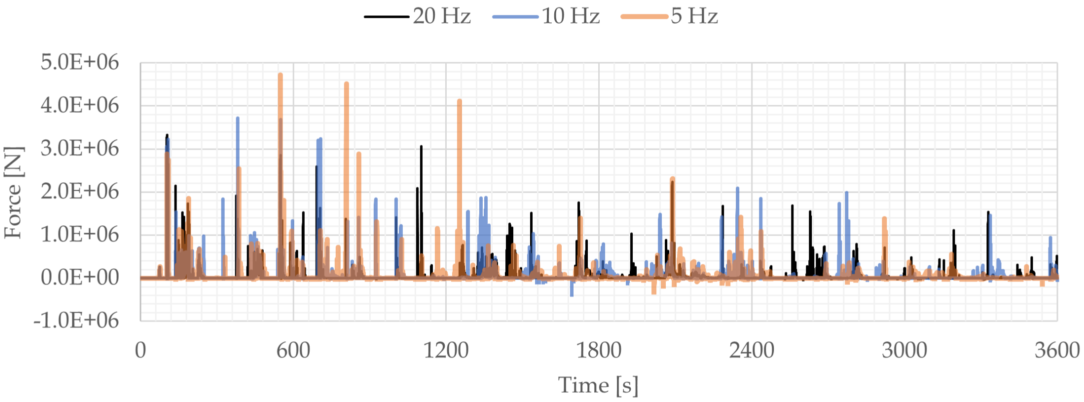


Figure 30: Ice forces on the floater in the X direction.

Frequency [Hz]	Ice force [N]			
	Min	Max	Mean	Std
5	-3.03E+05	4.72E+06	4.71E+04	1.64E+05
10	-3.73E+05	3.72E+06	4.87E+04	1.66E+05
20	-3.72E+04	3.06E+06	4.32E+04	1.30E+05

Table 8: Statistical summary of ice forces in the X direction

Visually, agreement between simulations can only be observed for the first collision on the first minute. After that, the structure experiences force bursts of different magnitude and at different times. With a higher frequency, less ice forces are simulated in the negative X direction, against the ice drift. As the frequency increases, the ice force maxima decrease. This can be explained by the fact that the ice force depends largely on the volume penetration of the structure into the ice. With a smaller timestep, it is less likely for extremely large penetrations to occur. The fact that the standard deviation decreases with increasing frequency potentially points to increased reliability of the simulation.

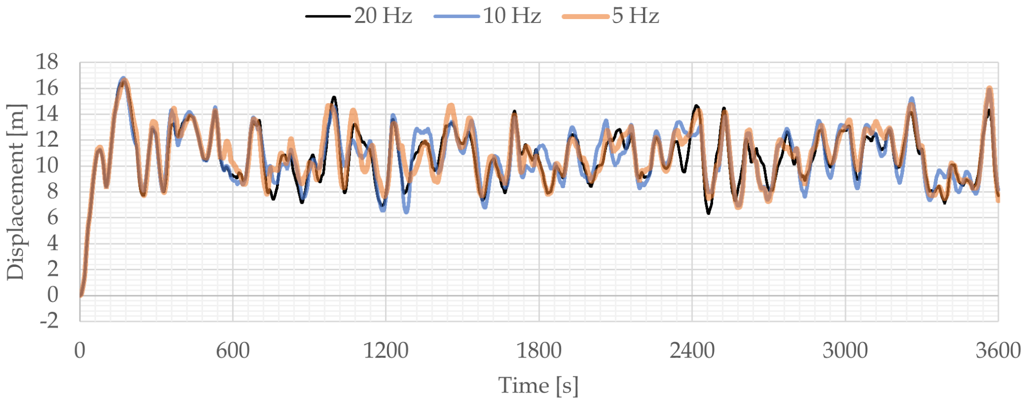


Figure 31: Displacement of the floater in the X direction.

Frequency [Hz]	Displacement [m]			
	Min	Max	Mean	Std
5	6.79E+00	1.61E+01	1.09E+01	1.85E+00
10	6.38E+00	1.59E+01	1.08E+01	1.90E+00
20	6.32E+00	1.54E+01	1.07E+01	1.82E+00

Table 9: Statistical summary of floater displacement in the X direction

As with the forces, the surge displacement is only modeled similarly for some 6 minutes, after which the first sign of divergence can be observed. The maximum and the mean displacement show reduction with increasing frequency. The highest frequency results in the lowest standard deviation, but it is not a steady decrease.

4.2.2 Sway

In sway, the simulation is shown to be completely dependent on simulation frequency. It is difficult to observe any pattern or sign of convergence in the force history. There is some

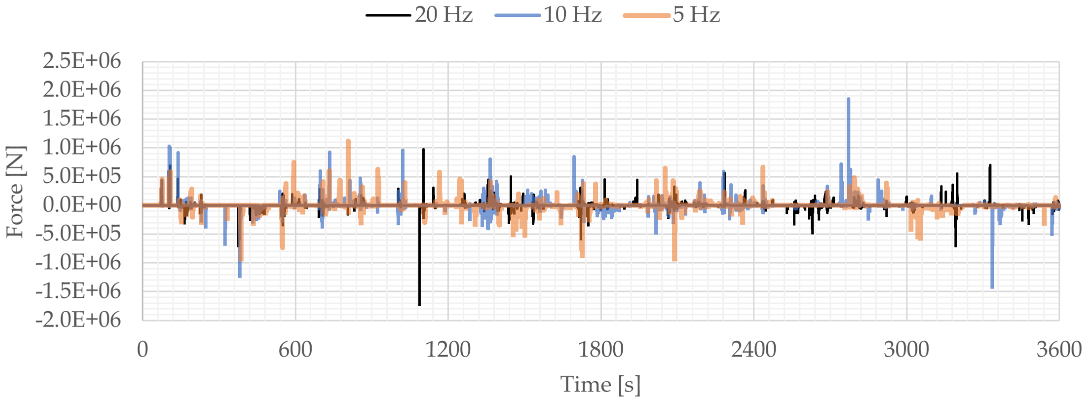


Figure 32: Ice forces on the floater in the Y direction.

Frequency [Hz]	Ice force [N]			
	Min	Max	Mean	Std
5	-9.19E+05	1.12E+06	2.48E+03	5.84E+04
10	-1.41E+06	1.85E+06	1.18E+03	5.34E+04
20	-1.72E+06	9.77E+05	1.10E+03	3.93E+04

Table 10: Statistical summary of ice forces in the Y direction

symmetry in the model around the global Y-axis. The mean sway force is expectedly low, one-two kilonewtons, for a structure of this mass. As the frequency increases, both the minima and the maxima shift closer to the mean.

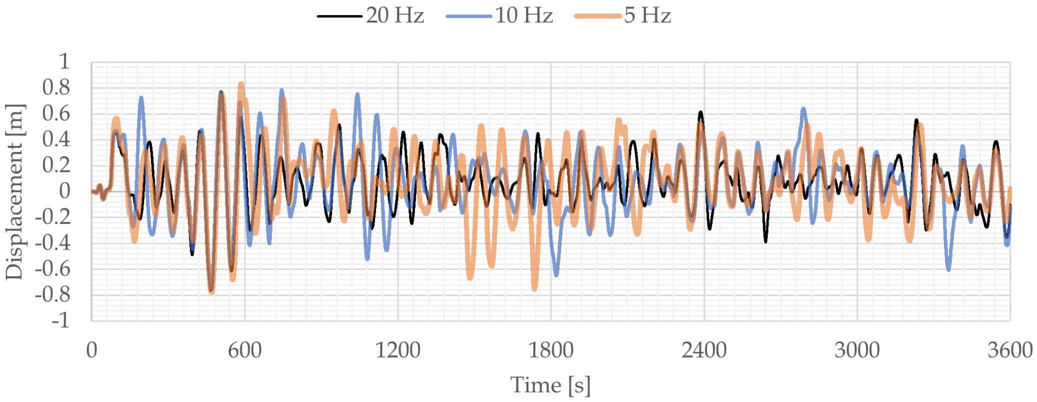


Figure 33: Displacement of the floater in the Y direction.

A weak pattern might be observed of oscillation amplitude decreasing with increasing simulation frequency. It is, however, a purely visual observation. It is somewhat supported by the tendency of the maximum and standard deviation of displacement to decrease with increasing frequency.

4.2.3 Heave

As in Section 4.1, 10-minute windows are presented starting from heave in Figure 34.

Frequency [Hz]	Displacement [m]			
	Min	Max	Mean	Std
5	-7.76E-01	8.35E-01	7.58E-02	2.67E-01
10	-7.39E-01	7.85E-01	6.95E-02	2.53E-01
20	-7.63E-01	7.72E-01	7.12E-02	2.06E-01

Table 11: Statistical summary of floater displacement in the Y direction

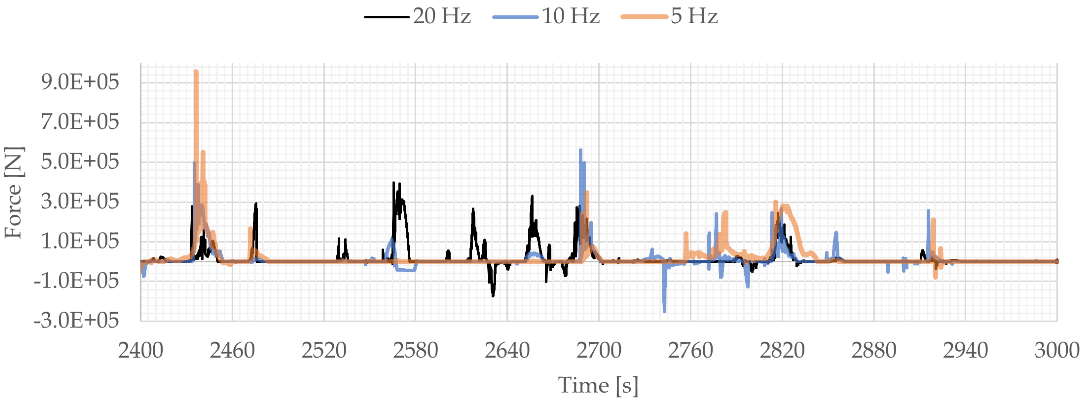


Figure 34: Ice forces on the floater in the Z direction.

Looking at ice force in heave, it is especially apparent that different timing of the collisions contributes significantly to the frequency dependence of the simulation. The absolute values of all statistical indicators decrease with increasing frequency. This suggests that a more accurate simulation should predict less significant vertical contributions from ice force.

From the heave displacement history, one could draw another conjectural conclusion that the larger divergence is observed after (differently timed) ice collisions. Conversely, given enough time without an ice collision, a behaviour will be simulated that is less dependent on frequency. The statistical summary of the heave displacement suggests that it is highly dependent on simulation frequency.

4.2.4 Roll

In Figure 36 through Figure 41, the ice torques around principal axes and corresponding displacements are presented. The torques are in the SIMA global frame, while the rotational displacements ϕ, θ, ψ are presented in the SIMA joint rotation system.

Frequency [Hz]	Ice force [N]			
	Min	Max	Mean	Std
5	-6.50E+05	9.57E+05	1.91E+04	5.16E+04
10	-4.81E+05	5.62E+05	1.35E+04	5.03E+04
20	-4.35E+05	6.52E+05	1.46E+04	4.18E+04

Table 12: Statistical summary of ice forces in the Z direction

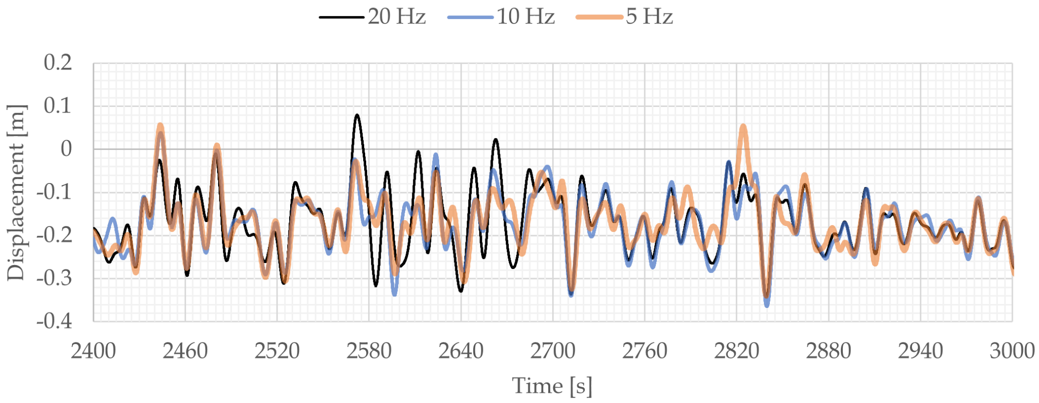


Figure 35: Displacement of the floater in the Z direction.

Frequency [Hz]	Displacement [m]			
	Min	Max	Mean	Std
5	-3.64E-01	5.73E-02	-1.69E-01	6.17E-02
10	-3.64E-01	3.86E-02	-1.69E-01	5.90E-02
20	-3.47E-01	8.03E-02	-1.68E-01	6.15E-02

Table 13: Statistical summary of floater displacement in the Z direction

There is still notable frequency-dependence of the ice forces. The roll forces show no sign of convergence.

The 10 Hz and 20 Hz roll histories are noticeably more similar than that of 5 Hz. This is the first sign of possible convergence, and simulation frequency should be further increased to confirm. The same pattern is observed in Figure 38 through Figure 41.

4.2.5 Pitch

Even though pitch should definitely not be expected to show any symmetry, the statistical summary still shows a trend towards a decreasing mean. The maximum torque is the only

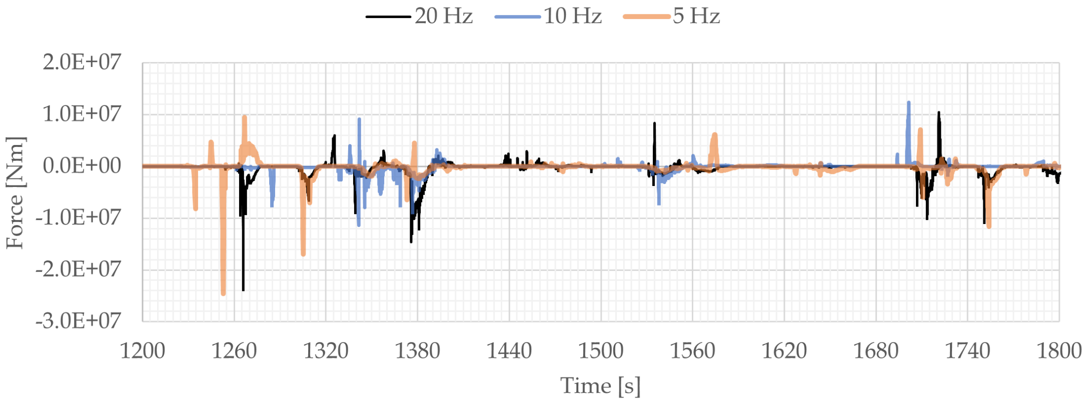


Figure 36: Ice forces on the floater in the ϕ direction.

Frequency [Hz]	Ice torque [N·m]			
	Min	Max	Mean	Std
5	-3.63E+07	1.52E+07	-1.04E+04	1.16E+06
10	-2.47E+07	1.44E+07	-1.38E+05	1.15E+06
20	-2.38E+07	1.60E+07	-1.01E+04	1.30E+06

Table 14: Statistical summary of ice torques around the X axis

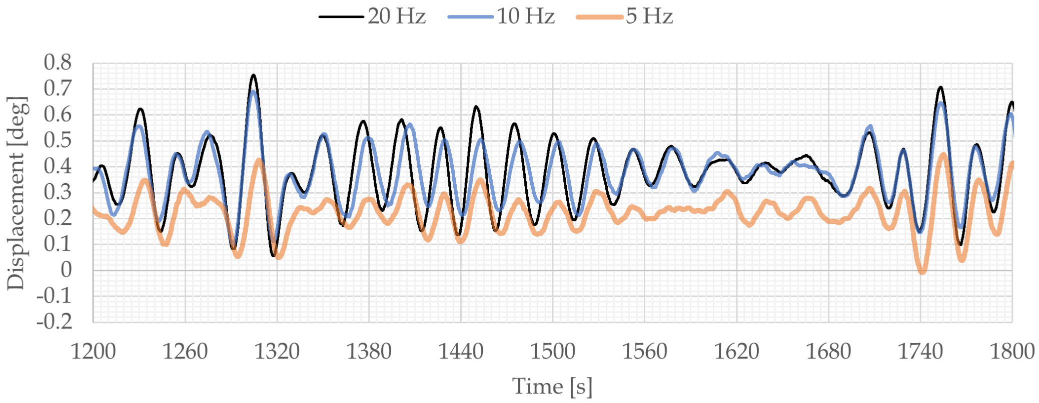


Figure 37: Displacement of the floater in the ϕ direction.

exception, with the lowest value simulated at 10 Hz.

Frequency [Hz]	Displacement [deg]			
	Min	Max	Mean	Std
5	-1.51E+00	9.46E+00	5.42E+00	2.18E+00
10	-1.97E+00	9.61E+00	5.36E+00	2.21E+00
20	-1.70E+00	9.70E+00	5.35E+00	2.19E+00

Table 17: Statistical summary of floater displacement in the θ direction

As the simulation frequency is increased, the pitch variation increases while the mean pitch decreases towards 5 degrees. The standard deviation does not change much.

Frequency [Hz]	Displacement [deg]			
	Min	Max	Mean	Std
5	-7.31E-03	5.18E-01	2.26E-01	7.58E-02
10	-1.70E-02	7.52E-01	3.74E-01	1.26E-01
20	9.54E-03	7.54E-01	3.73E-01	1.29E-01

Table 15: Statistical summary of floater displacement in the ϕ direction

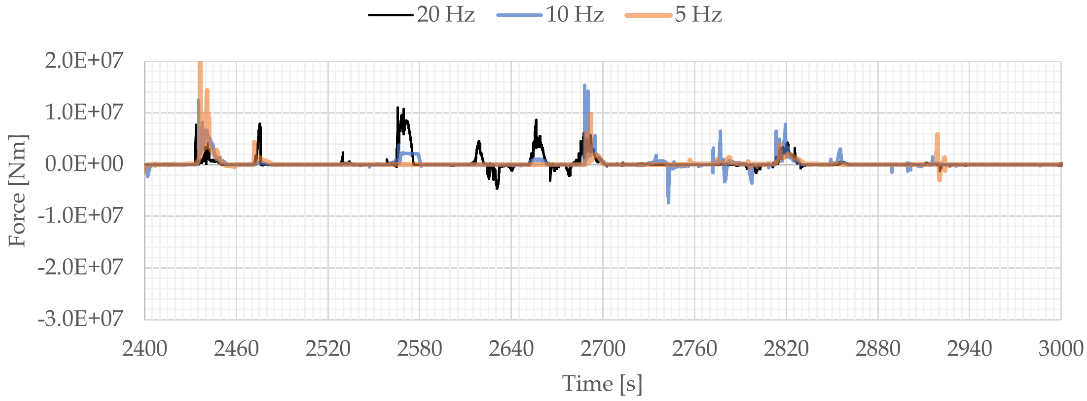


Figure 38: Ice forces on the floater in the θ direction.

Frequency [Hz]	Ice torque [N·m]			
	Min	Max	Mean	Std
5	-1.83E+07	2.44E+07	3.72E+05	1.19E+06
10	-1.34E+07	1.53E+07	3.11E+05	1.13E+06
20	-1.25E+07	1.71E+07	2.09E+05	9.53E+05

Table 16: Statistical summary of ice torques around the Y axis

4.2.6 Yaw

The yaw torque mean is highly frequency-dependent, with its mean value simulated on either side of zero in different cases. The standard deviation decreases as the frequency increases, pointing to some improvement.

It is much easier to assume convergence based on visual inspection of Figure 41, where the response changes much less as the frequency is increased beyond 10 Hz. The minima, maxima, and mean value of the yaw displacement stay relatively unchanged with increasing frequency. The lowest standard deviation is obtained with the highest frequency.

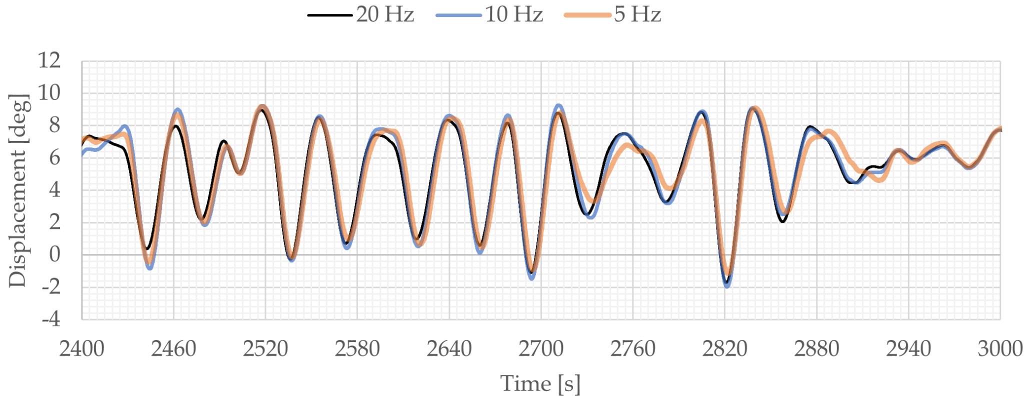


Figure 39: Displacement of the floater in the θ direction.

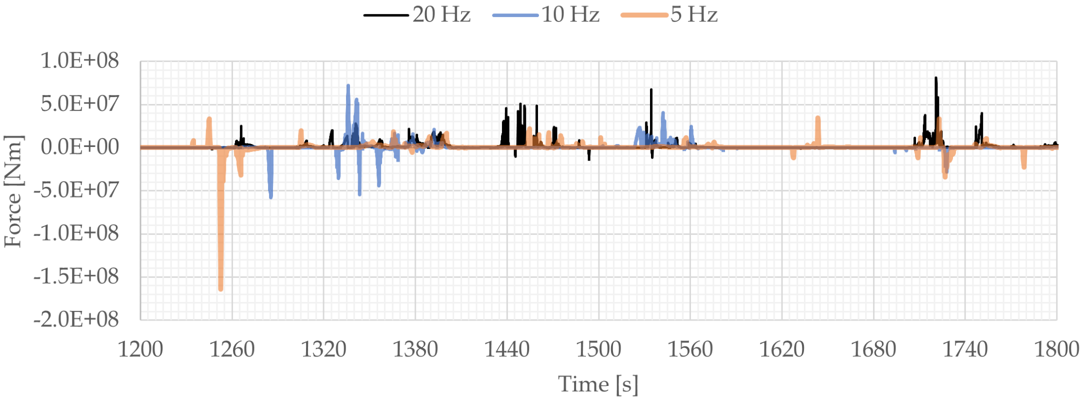


Figure 40: Ice forces on the floater in the ψ direction.

Frequency [Hz]	Ice torque [N·m]			
	Min	Max	Mean	Std
5	-2.05E+08	1.09E+08	-3.51E+05	6.69E+06
10	-1.46E+08	8.68E+07	1.07E+04	6.04E+06
20	-1.18E+08	1.18E+08	-3.69E+05	5.04E+06

Table 18: Statistical summary of ice torques around the Z axis

4.3 Influence of ice loads in comparison to other loads on the wind turbine

The presence or absence of other sea forces also influences the ice forces on the structure. Particularly, surge forces are of interest for mooring design. To illustrate the interaction between different forces on a wind turbine, several scenarios were simulated. The resulting ice forces are summarized in Table 20. Just as in Section 4.2, the first five minutes are excluded from the statistical summary to avoid capturing any transient effects. Time histories are also shown from simulations that were run with the SAMS coupling disabled, but otherwise with the same parameters. Thus, "coupled" in the legend means that the advancing ice field is included in the scenario. "Uncoupled" means that it is excluded.

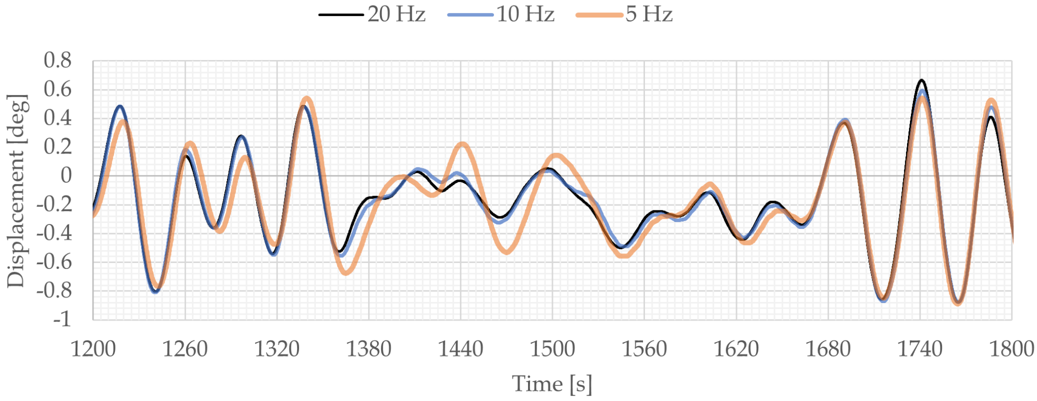


Figure 41: Displacement of the floater in the ψ direction.

Frequency [Hz]	Displacement [deg]			
	Min	Max	Mean	Std
5	-8.88E-01	6.11E-01	-1.70E-01	3.12E-01
10	-8.76E-01	7.02E-01	-1.67E-01	3.13E-01
20	-8.79E-01	6.66E-01	-1.71E-01	3.01E-01

Table 19: Statistical summary of floater displacement in the ψ direction

Scenario	Ice force[N]			
	Min	Max	Mean	Std
Calm sea without wind	0.00E+00	5.93E+05	4.95E+04	5.72E+04
Constant wind	-7.27E+03	2.77E+06	5.67E+04	1.30E+05
Turbulent wind	-2.04E+05	4.37E+06	5.78E+04	1.85E+05
Waves without wind	-2.23E+02	6.12E+05	5.02E+04	6.84E+04
Waves and constant wind	-1.68E+04	3.17E+06	5.82E+04	1.56E+05

Table 20: Statistical summary of longitudinal ice forces on a floater

While the mean value of the ice forces does not vary greatly between the scenarios, there are still some points of interest shown in following sections. The displacements due to the sum of all forces on the floater are summarized in Table 21.

Scenario	Displacement [m]			
	Min	Max	Mean	Std
Calm sea without wind	2.63E-01	2.82E+00	1.04E+00	4.59E-01
Constant wind	1.19E+01	1.88E+01	1.36E+01	8.93E-01
Turbulent wind	7.01E+00	1.51E+01	1.08E+01	1.79E+00
Waves without wind	1.62E-01	3.27E+00	1.05E+00	4.97E-01
Waves and constant wind	1.14E+01	1.75E+01	1.37E+01	8.36E-01

Table 21: Statistical summary of longitudinal displacements of a floater

The mean longitudinal displacement is always positive. This is expected, because even in the absence of wind thrust on the turbine, there is still always a current in the sea driving the ice field. This current also exerts a force on the floater and mooring. As somewhat expected, the lowest mean forces and displacements are encountered in calm sea conditions. This suggests that for design scenarios, coupled analysis is more useful to identify potential extreme loads.

In addition to the scenarios studied here, a case with waves and turbulent wind was shown in Section 4.2.

4.3.1 Calm sea

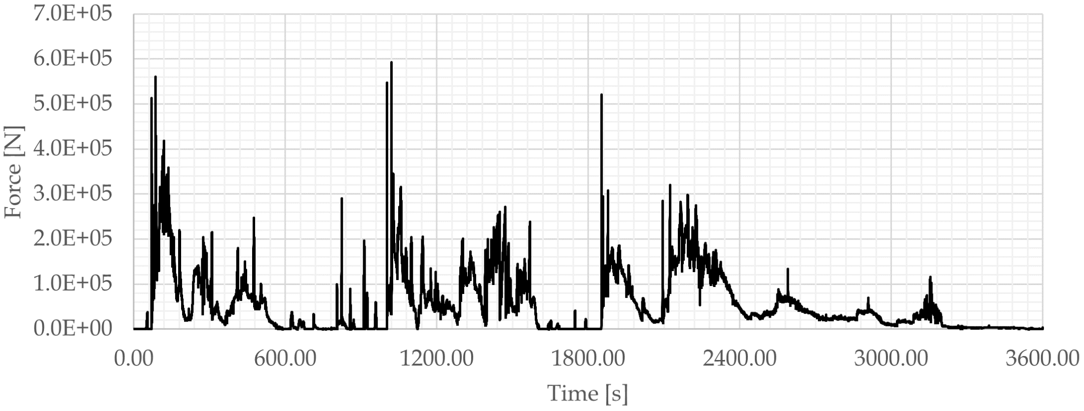


Figure 42: Longitudinal ice force on the structure in calm sea.

This is the only scenario where the structure never experiences negative forces. In all other scenarios, due to stochastic movement of the floater, sometimes the ice contact results in a force directed against the oncoming ice. The calm sea scenario is also characterized by long periods of ice contact. The implications of prolonged ice forces were discussed in Section 4.1.

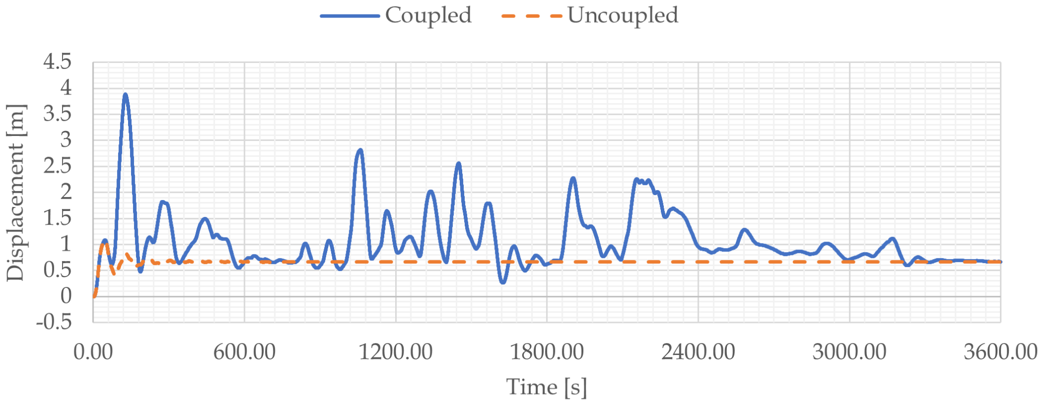


Figure 43: Longitudinal displacement of the structure in calm sea.

The displacements can be clearly identified with the forces that cause them. It can be seen that even though there are momentary spikes of extremely high ice force, they are so short that the actual momentum transfer to the structure is not great. Long periods of mean-value ice forces contribute more to displacement.

4.3.2 Constant wind

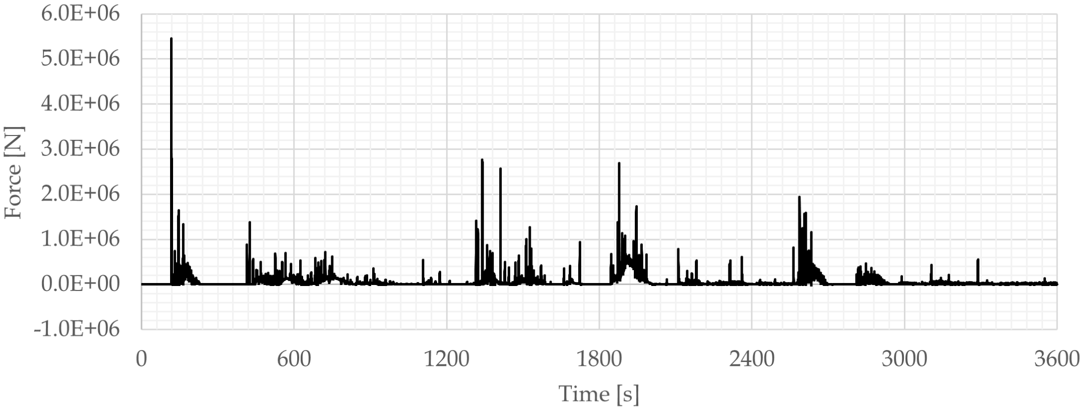


Figure 44: Longitudinal ice force on the structure with constant wind.

In constant wind conditions, the structure experiences ice forces that at their extreme are one order of magnitude larger than in calm sea. The most obvious explanation is that the pitch angle due to wind thrust is large. Because of it, the cones move away from the waterline and do not perform their intended function anymore. This quite common occurrence is illustrated in Figure 45.

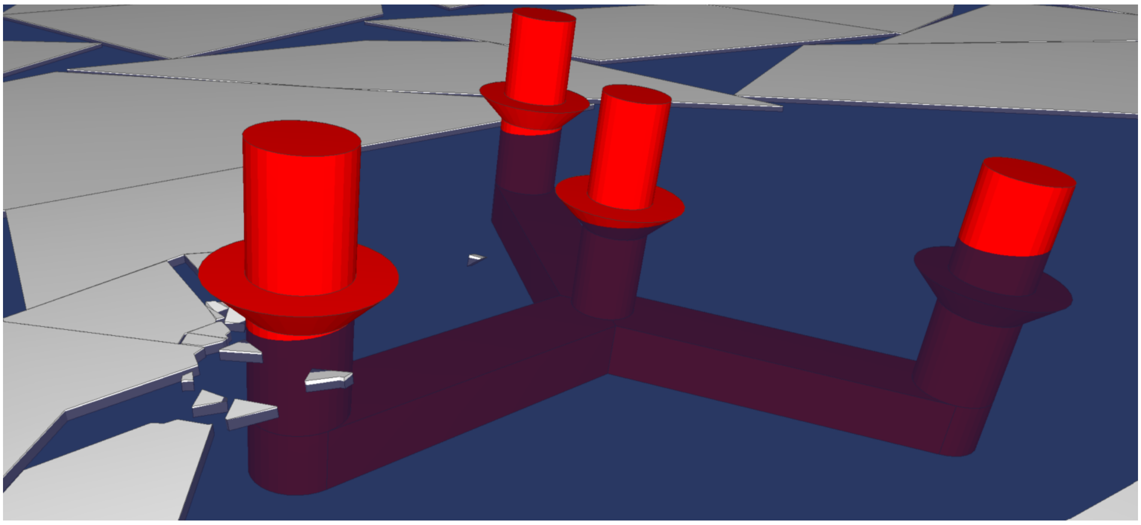


Figure 45: Submergence and emergence of ice cones due to wind-induced pitch.

Visualizations show that the when the pitch varies due to turbulent wind, the cones rise in and out of water, breaking the ice like a hammer. This is not, however, the intention behind their design. This result points to a unsuccessful choice of ice cone geometry for this thesis.

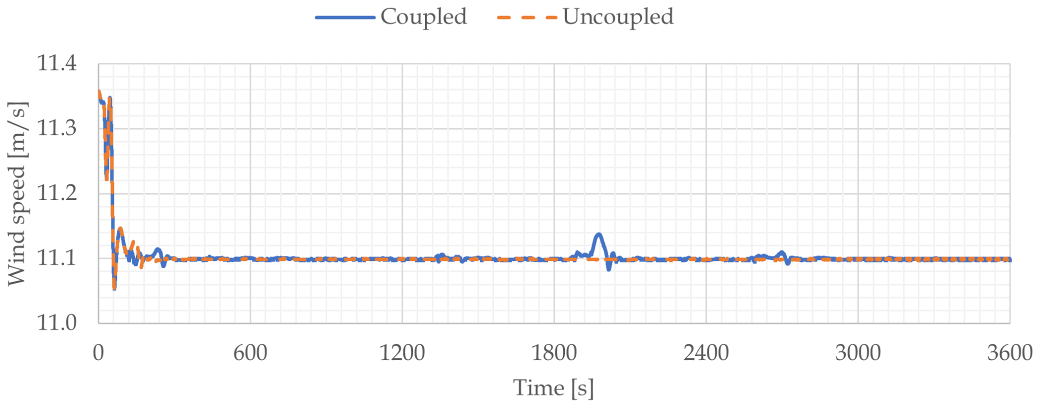


Figure 46: Incoming longitudinal wind speed on the turbine shaft.

An interesting observation is that in the absence of wind, the ice collisions can be detected by the relative wind speed at the wind turbine shaft. This only happens if the ice contact is long enough to transfer a significant amount of momentum to the floater, thus causing a large displacement.

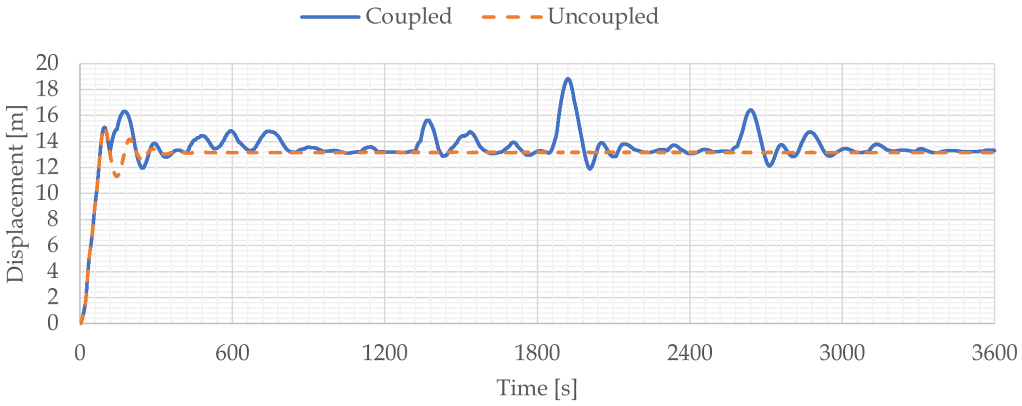


Figure 47: Longitudinal displacement of the structure with constant wind.

As expected, the relative wind gusts at the nacelle coincide with the greatest displacements of the floater. The measured wind speed also depends greatly on the pitch response, which is not showed here.

4.3.3 Turbulent wind

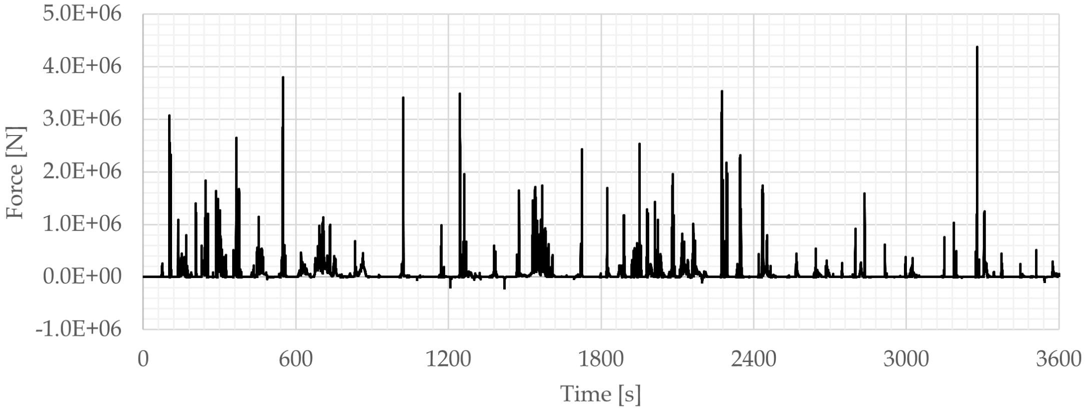


Figure 48: Longitudinal ice force on the structure with turbulent wind.

This scenario exhibits the greatest range of extreme ice forces, both negative and positive. The variation of the ice forces is the largest in this scenario, as well. The standard deviation is approaching 2 m. There are very few prolonged ice contact periods due to the erratic movement of the structure.

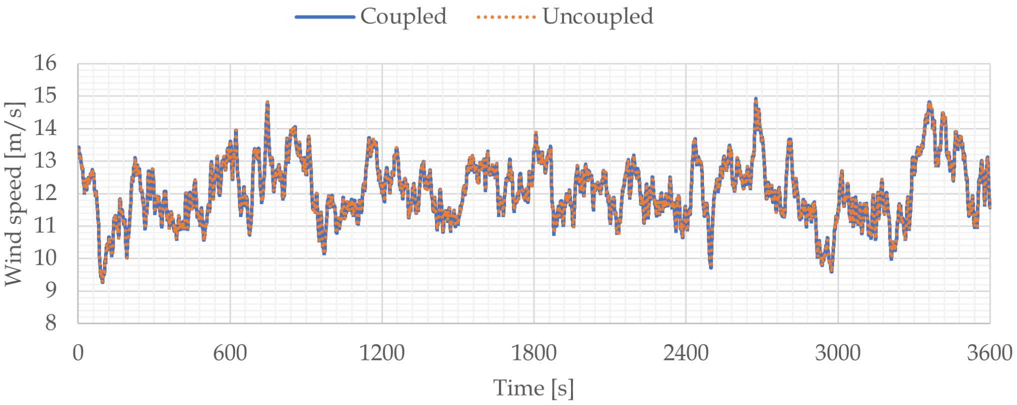


Figure 49: Incoming longitudinal wind speed on the turbine shaft.

Even though there might be some influence on the measured wind speed from ice collisions, it is not distinguishable from the large variation caused by turbulence itself.

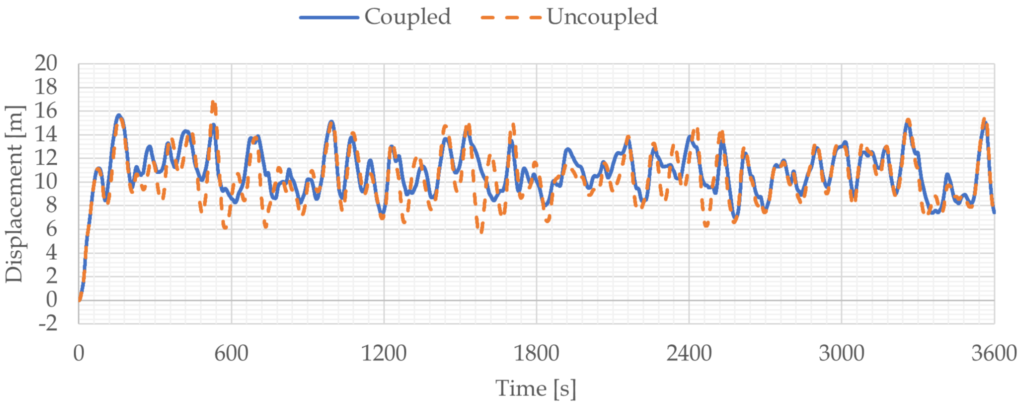


Figure 50: Longitudinal displacement of the structure with turbulent wind.

Although it can not be claimed with full confidence, the uncoupled displacement history seems to generally have a larger amplitude than the coupled history. This is purely a visual observation. If true, it could be explained by the fact that the mean surge force from ice causes a mean displacement in surge. This, in turn, increases the mooring line tension and the total stiffness of the system in surge. Thus, the response is shifted out of the excitation frequency range of the turbulent wind.

4.3.4 Waves

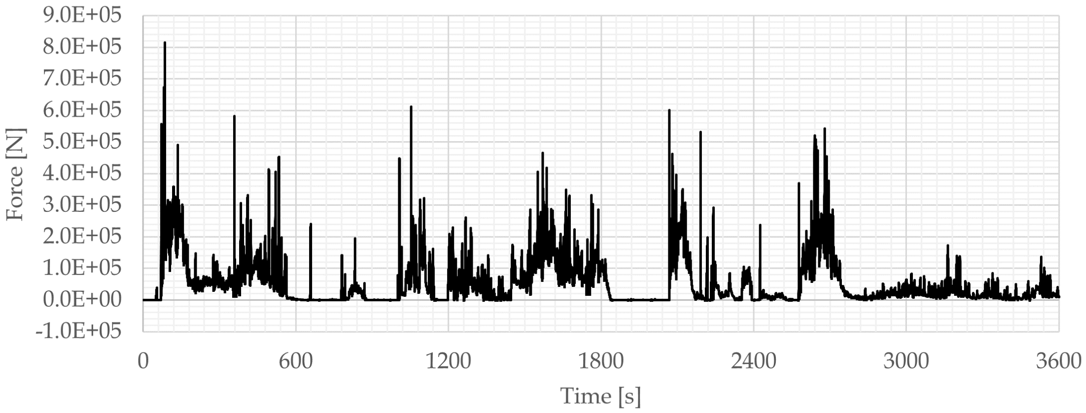


Figure 51: Longitudinal ice force on the structure in sea waves.

The ice force history in the case with waves, but without wind, is quite similar to the calm sea case. The only noticeable difference is the presence of negative ice forces, caused by the movement of the floater due to wave loads.

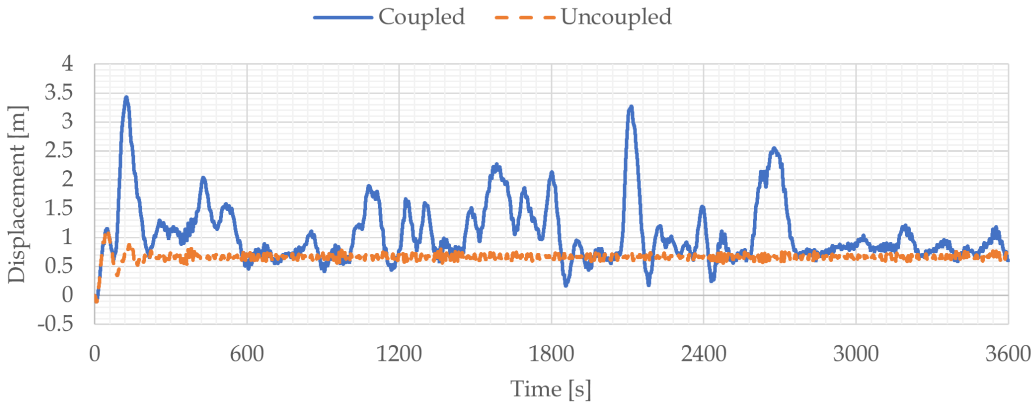


Figure 52: Longitudinal displacement of the structure in sea waves.

As they were chosen for this scenario, the massive floes with a thickness of 1 m cause much larger displacements than waves with a significant height of 0.3 m. Since the choice was arbitrary, not much can be concluded from the comparison. To make a more meaningful comparison, joint wave and ice conditions should be chosen according to statistics of field measurements in the sea.

4.3.5 Waves and constant wind

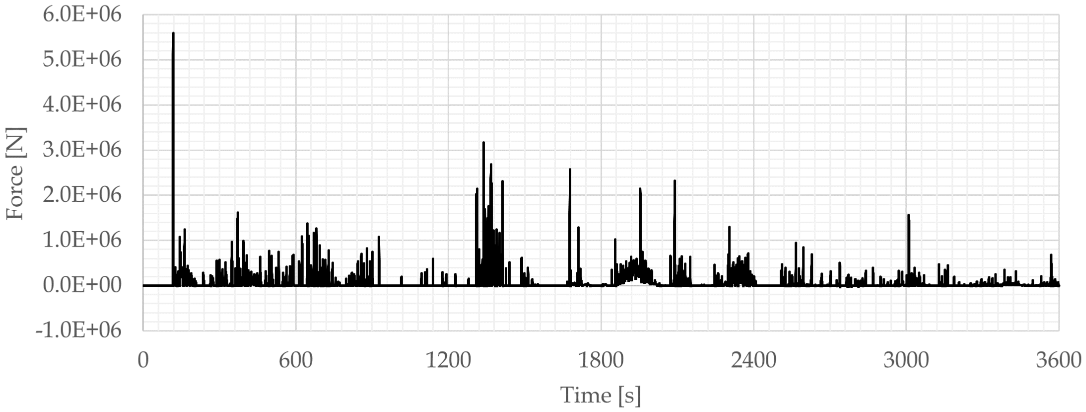


Figure 53: Longitudinal ice force on the structure with sea waves and constant wind.

The largest mean ice force is observed in this scenario. There is an ice contact period of some minutes, but generally the wave motions seem to prevent lengthy ice interactions. Instead, the ice force is exerted in short bursts.

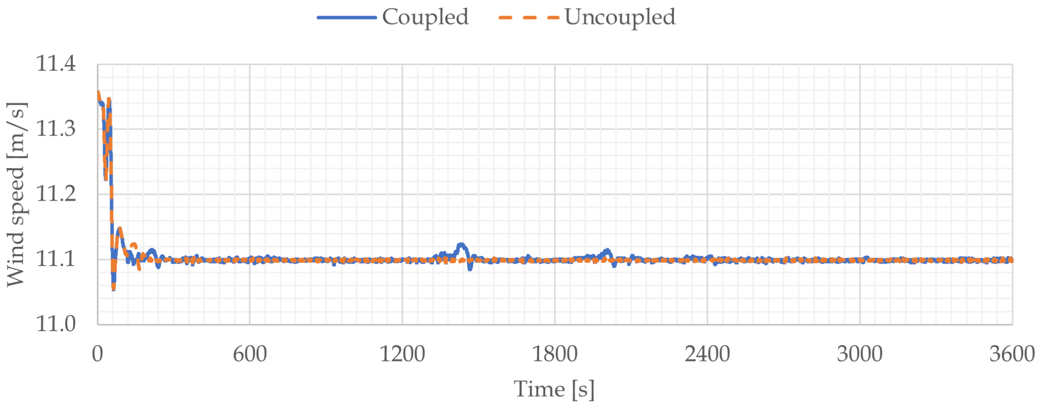


Figure 54: Incoming longitudinal wind speed on the turbine shaft.

Just as without waves, the wind variation is small enough that ice contact events can be detected from wind speed measurement.

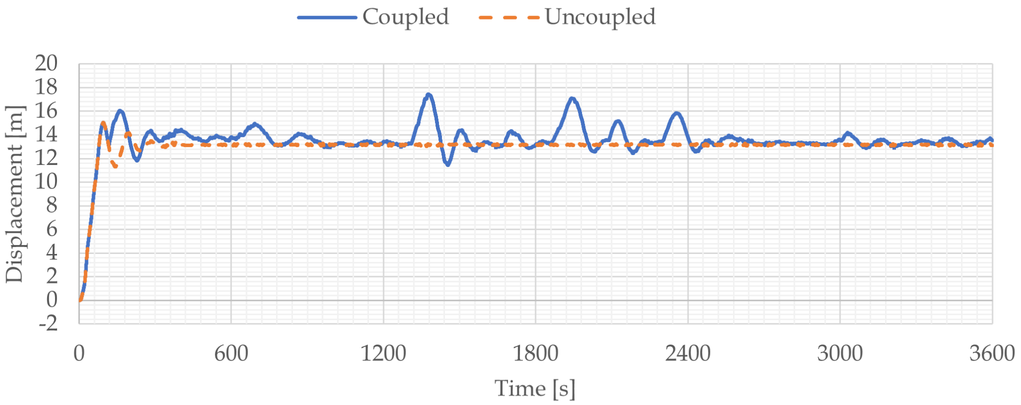


Figure 55: Longitudinal displacement of the structure with sea waves and constant wind.

By its statistical summary, the displacement history of this scenario is very similar to the constant wind scenario without waves. This is yet another clue to the fact that perhaps the significant wave height was chosen too low for this study, or conversely the ice too thick.

5 Discussion

5.1 Timestep sensitivity

The coupling model shows a definite sensitivity to the simulation frequency. All environmental inputs being the same, a different frequency causes ice fractures to happen differently and at different times. These deviation propagate through the simulation. After one hour, the final state of the simulation is visibly different.

It is not certain whether there is a high enough simulation frequency at which the behaviour of the system will become independent of further frequency increase. It could be possible that with a small timestep, the ice floe penetration will increase gradually enough from one timestep to the next. This in turn will cause the fracture to be calculated at one certain time and location. This could then be considered the "true" solution.

Such a timestep-independent solution was not reached in this study. The prohibitive computation time growth may be a serious obstacle to finding one. Thus, one approach would be to increase the speed of the simulation. For example, deleting small ice floes after a certain reduction in size or distance travelled past the structure might help with limiting the amount of bodies in the simulation. This method has been successfully used in Rolandsen and Hoel, 2018 and van den Berg et al., 2019.

5.2 Ice forces

The coupling code was successful in transferring the ice forces from SAMS to SIMA. The position and orientation of the rigid floater is reliably followed in SAMS, even during prolonged ice contact periods. There is a delay Δt in this following. However, Δt is small compared to the physical processes that the model is trying to capture. For this reason, the effect of the delay can be neglected.

The quantitative analysis of the ice forces doesn't have a significant meaning beyond confirmation that the coupling code works. The environmental conditions were arbitrarily chosen. They do not correspond to a valid design scenario. For this reason, one shouldn't draw too many conclusions from the fact that the wind caused larger displacements than the ice in this thesis, or that the ice-induced surge displacements were far greater than those induced by the waves.

Comparing the ice force statistics in Section 4.3 with Eik and Aksnes, 2010, it can be seen than the max and mean loads in this thesis are not related by a ratio. Instead, the extreme loads exceed the mean loads by orders of magnitude. This can be interpreted as further confirmation that broken ice loads on moored structures are harder to predict than level ice loads. Thus, further study in this area is well motivated.

5.3 Cone design

The ice-breaking cone geometry is another arbitrary choice made during this study that proved to be wrong. Due to its large expected pitch response, a semi-submersible platform for a wind turbine should have ice protection features that function over a large vertical section of the ice-facing area. One option is to use a steeper slope angle for the cones, to cover more vertical distance while preserving the largest diameter. Another option is to use a downward-sloping structure below the waterline and an upward-sloping structure above it. In such a case, one should study the interaction of the wedge-shaped outer diameter and the ice floes.

6 Conclusion

In order for the world to meet its climate goals, wind energy must be installed at increasing paces. Floating wind turbines have great potential to contribute to the worldwide wind energy capacity. Their main advantage, space versatility, can be greatly enhanced by considering cold regions for installation. A challenge that comes with cold regions is floating sea ice. It is well-justified to study the interactions between sea ice and the semi-submersible floater of a wind turbine.

Closed-form methods exist for calculating ice forces on marine structure, both fixed and floating. however, they should be supported by time-domain analysis methods. Several of these exist in literature today, but none that permit analysis of a floating, moored, elastic, pitch-controlled wind turbine in coupled sea and ice conditions.

Such an analysis can be performed in cooperation between two programs, SIMA and SAMS. SIMA performs the simulation of the elastic structure and pitch-controlled wind turbine in sea conditions. SAMS performs the many-body simulation with individual ice floes and collision contact. The connection between them was the contribution of this thesis, made possible due to the DLL interface available in SIMA and TCP interface available in SAMS. The code to facilitate communication was written in C and is executed on every timestep of the simulation.

To make sure that both programs simulate the same scenario, a coupling algorithm was developed. In this algorithm, a coupling force $q_{coupling}$ was derived. Applied as an external force to the structure in SAMS, it ensures that the structure in SAMS follows the structure in SIMA exactly. Thus, collision events are predicted and the ice contact forces are sent back to SAMS. In this communication, special attention was paid to the treatment of coordinate frames and orientations of structures in space. Thus, the difference in definitions of principal axes in the two programs was overcome.

A simulation was set up to test different aspects of the coupling code. The ice, wind, and wave conditions were varied in different combinations. Additionally, the simulation frequency was varied to find whether the simulation method exhibits any timestep dependence.

The coupling code was found to be robust and ensure identical behaviour of structures in the two programs, even during prolonged ice contact periods. The method was, however, found highly dependent on simulation frequency. The exact timing of the ice contact events, as well as the forces involved and subsequent trajectory of the bodies involved, all varied to an increasing degree as the simulation progressed. By the end of the hour-long simulation, the final state of the simulated scenario was visibly different depending on the simulation frequency. Some weak evidence of convergence was found, but ultimately not enough to claim that further increase of the simulation frequency would solve the problem. Increasing the frequency, as it was found, is computationally expensive.

The structure geometry had been augmented with ice-breaking cones. Their design proved ineffective for the sample scenario simulated in this thesis. Due to wind-induced pitch, the ice cones moved away from the waterline. Instead, the advancing ice field made contact with the vertical cylindrical column surfaces of the semi-submersible, likely leading to increased ice forces on the structure.

6.1 Future work

The author sees potential future work in two directions: using the code developed in this scenario for engineering purposes, or improving the code itself.

The code is robust enough to permit simulating a complex marine structure in broken ice. Even with the limitations it has now, some useful conclusions could be drawn from simulations performed with this code. Validation studies should take priority, with scenarios matching either laboratory, towing tank, or field tests.

Efforts to improve the code may go towards user friendliness and versatility, ensuring that an uninitiated engineer is able to use the code without too much of a struggle. Even more useful, perhaps, would be speed improvements. One such improvement, deletion of ice floes that are no longer relevant, has been proposed in the discussion of results of this thesis. A more complicated challenge would be to tackle the severe timestep dependence of the code, requiring a thorough re-evaluation of the working principles behind the algorithm and the programs it is connecting.

Bibliography

Afanasjev, M. (2021). Ice loads on floating wind turbines.

Amdahl, T. H., Bjørnsen, B., Hagen, O. J. & Søberg, S. R. (2014). Numerical simulations of ice loads on an Arctic Floater in managed ice. https://doi.org/10.4043/24549-MS

ArcISO. (2020). Text based user control features of SAMS.

Bachynski, E. E. (2014). *Design and Dynamic Analysis of Tension Leg Platform Wind Turbines* [Accepted: 2014-12-19T12:10:49Z]. Norges teknisk-naturvitenskapelige universitet, Fakultet for ingeniørvitenskap og teknologi, Institutt for marin teknikk. Retrieved 9th June 2022, from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/238768

Bak, C., Zahle, F., Bitsche, R., Taeseong, K., Yde, A., Henriksen, L. C., Hansen, M. H., Blasques, J. P. A. A., Gaunaa, M. & Natarajan, A. (2013). The DTU 10-MW reference wind turbine. *Dainsh wind power research.*

Barker, A., Timco, G. & Wright, B. (2000). Numerical Simulation of the "Kulluk" in Pack Ice Conditions. *Proceedings of the 15th IAHR International Symposium on Ice*, *1*.

Bilgili, M., Yasar, A. & Simsek, E. (2011). Offshore wind power development in Europe and its comparison with onshore counterpart. *Renewable and Sustainable Energy Reviews*, *15*(2), 905–915. https://doi.org/10.1016/j.rser.2010.11.006

Bonnemaire, B., Serré, N., Lundamo, T., Fredriksen, A., Jensen, A., Gürtner, A. & Teigen, S. (2014). Ice Breaking and Accumulation Around a Moored Structure: Ice Basin Tests and Numerical Simulations. https://doi.org/10.4043/24579-MS

Bonnemaire, B., Tan, X., Serré, N., Fredriksen, A., Metrikin, I. & Gürtner, A. (2015). Post-simulations of Ice Basin Tests of a Moored Structure in Broken Ice - Challenges and Solutions. https://doi.org/10.4043/25531-MS

Croasdale, K. R., Bruce, J. R. & Liferov, P. (2009). Sea Ice Loads Due to Managed Ice [ISSN: 0376-6756 Number: POAC09-31]. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions.* Retrieved 25th January 2022, from https://trid.trb.org/view/1342210

DNV. (2021a). *Energy Transition Outlook 2021 - Pathway to Net Zero Emissions* (tech. rep.). Høvik, Norway.

DNV. (2021b). *Energy Transition Outlook 2021 - Technology Progress Report* (tech. rep.). Høvik, Norway.

DNV. (2021c). Marine operations and mooring analysis software — Sima. Retrieved 10th June 2022, from https://www.dnv.com/services/marine-operations-and-mooring-analysis-software-sima-2324

DTU. (2021). Global Wind Atlas. Retrieved 11th June 2022, from https://globalwindatlas.info

Eik, K. (2011). Sea-ice management and its impact on the design of offshore structures. *Cold Regions Science and Technology*, *65*(2), 172–183. https://doi.org/10.1016/j.coldregions.2010.10.009

Eik, K. & Aksnes, V. (2010). Characterisation of Peak Loads on a Moored Production Vessel in Ice.

Equinor. (2021). Hywind Scotland remains the UK's best performing offshore wind farm. https://www.equinor.com/en/news/20210323hywind-scotland-uk-best-performing-offshore-wind-farm.html

Global CCS Institute. (2021). Accelerating to Net Zero.

Hansen, E. & Løset, S. (1999a). Modelling floating offshore units moored in broken ice: Comparing simulations with ice tank tests. *Cold Regions Science and Technology*, *29*(2), 107–119. https://doi.org/10.1016/S0165-232X(99)00017-8

Hansen, E. & Løset, S. (1999b). Modelling floating offshore units moored in broken ice: Model description. *Cold Regions Science and Technology*, *29*(2), 97–106. https://doi.org/10.1016/S0165-232X(99)00023-3

Hansen, M. (2015). *Aerodynamics of wind turbines.*

IEA. (2020). Energy Technology Perspectives.

Intel. (2021). Compile Cross-Architecture: Intel® oneAPI DPC++/C++ Compiler. Retrieved 10th June 2022, from https://www.intel.com/content/www/us/en/developer/tools/oneapi/dpc-compiler.html

Islam, M., Mills, J., Gash, R. & Pearson, W. (2021). A literature survey of broken ice-structure interaction modelling methods for ships and offshore platforms. *Ocean Engineering*, *221*, 108527. https://doi.org/10.1016/j.oceaneng.2020.108527

Karulin, E. B. & Karulina, M. M. (2010). Performance Studies For Technological Complex Platform Prirazlomnaya - Moored Tanker In Ice Conditions. Retrieved 26th January 2022, from https://onepetro.org/ISOPEPACOMS/proceedings/PACOMS10/All-PACOMS10/ISOPE-P-10-005/25826

Karulin, E. B. & Karulina, M. M. (2011). Numerical and physical simulations of moored tanker behaviour [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17445302.2010.544087 *Ships and Offshore Structures*, *6*(3), 179–184. https://doi.org/10.1080/17445302.2010.544087

Karulin, E. B. & Karulina, M. M. (2013). Determination of Loads on Mooring System During the Semisubmersible Interaction With Ice. Retrieved 26th January 2022, from https://onepetro.org/ISOPEIOPEC/proceedings/ISOPE13/All-ISOPE13/ISOPE-I-13-167/15162

Karulin, E. B. & Karulina, M. M. (2014). Peculiarities of Multi-Legged Platform Operation in Ice Condition. https://doi.org/10.1115/OMAE2014-23203

Langen, I. & Sigbjörnsson, R. (1979). *Dynamisk analyse av konstruksjoner.* Tapir.

Lau, M. (2006). Discrete element modeling of ship maneuvering in ice. *18th IAHR International Symposium on Ice*, *2*, 25–32.

Lau, M., Lawrence, K. P. & Rothenburg, L. (2011). Discrete element analysis of ice loads on ships and structures [Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17445302.2010.544 *Ships and Offshore Structures*, *6*(3), 211–221. https://doi.org/10.1080/17445302.2010.544086

Liferov, P. (2014). Station-Keeping in Ice – Normative Requirements and Informative Solutions. https://doi.org/10.4043/24580-MS

Liu, J. J., Liu, X., Chen, Y., Long, X. & Ji, S. (2016). Numerical Simulations of Ice Loads on Fixed and Floating Offshore Structures using the Discrete Element Method. https://doi.org/10.4043/27414-MS

Lu, W., Lubbad, R. & Løset, S. (2015a). In-plane fracture of an ice floe: A theoretical study on the splitting failure mode. *Cold Regions Science and Technology*, *110*, 77–101. https://doi.org/10.1016/j.coldregions.2014.11.007

Lu, W., Lubbad, R. & Løset, S. (2015b). Out-of-plane failure of an ice floe: Radial-crack-initiation-controlled fracture. *Cold Regions Science and Technology*, *119*, 183–203. https://doi.org/10.1016/j.coldregions.2015.08.009

Lu, W., Lubbad, R., Løset, S. & Kashafutdinov, M. (2016). Fracture of an ice floe: Local out-of-plane flexural failures versus global in-plane splitting failure. *Cold Regions Science and Technology*, *123*, 1–13. https://doi.org/10.1016/j.coldregions.2015.11.010

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. https://doi.org/10.1016/j.coldregions.2010.09.004

Lubbad, R., Løset, S., Lu, W., Tsarau, A. & van den Berg, M. (2018). Simulator for Arctic Marine Structures (SAMS). https://doi.org/10.1115/OMAE2018-78592

Lubbad, R., Lu, W., van den Berg, M., Løset, S. & Tsarau, A. (2020). Advances in Simulator for Arctic Marine Structures. Retrieved 25th May 2022, from https://onepetro.org/ISOPEIOPEC/proceedings/ISOPE20/All-ISOPE20/ISOPE-I-20-1243/446405

Marintek. (2009). SIMO - User's Manual.

Metrikin, I. (2014). A Software Framework for Simulating Stationkeeping of a Vessel in Discontinuous Ice. *Modeling, Identification and Control: A Norwegian Research Bulletin*, *35*(4), 211–248. https://doi.org/10.4173/mic.2014.4.2

Metrikin, I., Gürtner, A., Bonnemaire, B., Tan, X., Fredriksen, A. & Sapelnikov, D. (2015). SIBIS: A Numerical Environment for Simulating Offshore Operations in Discontinuous Ice [ISSN: 0376-6756]. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*. Retrieved 26th January 2022, from https://trid.trb.org/view/1429255

Metrikin, I., Teigen, S. H., Gürtner, A., Uthaug, E. S., Sapelnikov, D., Ervik, Å., Fredriksen, A., Lundamo, T. & Bonnemaire, B. (2015). Experimental and Numerical Investigations of a Ship-Shaped, Turret-Moored Floating Structure in Intact and Managed Sea Ice Conditions. https://doi.org/10.4043/25528-MS

Murray, J. J. & Spencer, D. S. (1997). A simulation model for a turret moored tanker in pack ice cover. *4*, 127–140.

Musial, W., Spitsen, P., Beiter, P., Duffy, P., Marquis, M., Cooperman, A., Hammond, R. & Shields, M. (2021). Offshore Wind Market Report: 2021 Edition, 119.

Nicolas, S., Sofien, K., Charles, P., Matthias, R. & Dmitry, S. (2019). Numerical Simulation of Broken Ice Interaction With Offshore Structures: Validation Exercises [ISSN: 0376-6756]. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*. Retrieved 26th January 2022, from https://trid.trb.org/view/1717503

O'Reilly, O. M. (2020). *Intermediate Dynamics for Engineers: Newton-Euler and Lagrangian Mechanics* (2nd ed.). Cambridge University Press. https://doi.org/10.1017/9781108644297

Palmer, A. & Croasdale, K. (2012). *Arctic offshore engineering* [Book Title: Arctic offshore engineering]. World Scientific Publishing CoPteLtd. https://doi.org/10.1142/8283

Pettineo, E. (2021). *Dynamic response analysis of a semi-submersible floating wind turbines during combined ice and aerodynamic loads* (Master's thesis in Maritime Engineering). NTNU. Trondheim.

Richard, M. & McKenna, R. (2013). Factors Influencing Managed Sea Ice Loads [ISSN: 0376-6756]. *Proceedings of the International Conference on Port and Ocean Engineering Under Arctic Conditions*. Retrieved 26th January 2022, from https://trid.trb.org/view/1328266

Rodrigues, O. (1840). Des lois géometriques qui regissent les déplacements d' un systéme solide dans l' espace, et de la variation des coordonnées provenant de ces déplacement considérées indépendant des causes qui peuvent les produire. *J. Math. Pures Appl.*, *5*, 380–440.

Rolandsen, A. N. & Hoel, P. J. (2018). Digital Twin of Vessels in Arctic Environments - Extending a Simulation Environment to allow for External Control of Multiple Vessels [Accepted: 2018-09-25T14:03:39Z Publisher: NTNU]. Retrieved 9th February 2022, from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2564514

Sayed, M. & Barker, A. (2011). Numerical Simulations of Ice Interaction with a Moored Structure. https://doi.org/10.4043/22101-MS

Sayed, M., Islam, S., Watson, D. & Wright, B. (2018). An Illustration of Acceptable Ice Conditions for Vessel Station-Keeping Operations in Pack Ice. Retrieved 27th January 2022, from https://onepetro.org/ISOPEIOPEC/proceedings/ISOPE18/All-ISOPE18/ISOPE-I-18-575/20167

Sayed, M., Kubat, I., Watson, D., Wright, B., Gash, R. & Millan, J. (2015). Simulations of the StationKeeping of Drillships Under Changing Direction of Ice Movement. https://doi.org/10.4043/25565-MS

Sayed, M., Kubat, I., Wright, B. & Millan, J. (2014). Numerical Simulations of Ice Forces on Moored and Thruster-Assisted Drillships. https://doi.org/10.4043/24647-MS

Sayed, M., Kubat, I. K. & Wright, B. (2012). Numerical Simulations of ice forces on the Kulluk: The role of ice confinement, ice pressure and ice management. https://doi.org/10.4043/23823-MS

Sayed, M., Watson, D., Kubat, I. & Wright, B. (2016). Simulations of the Effect of the Rate of Change of Ice Direction on Stationkeeping of Drillships. https://doi.org/10.4043/27427-MS

Shields, M., Beiter, P., Nunemaker, J., Cooperman, A. & Duffy, P. (2021). Impacts of turbine and plant upsizing on the levelized cost of energy for offshore wind. *Applied Energy*, *298*, 117189. https://doi.org/10.1016/j.apenergy.2021.117189

SINTEF, O. (2018). SIMO 4.14.0 Theory Manual.

Su, B., Aarsæther, K. G. & Kristiansen, D. (2019). Numerical Study of a Moored Structure in Moving Broken Ice Driven by Current and Wave. *Journal of Offshore Mechanics and Arctic Engineering*, *141*(3). https://doi.org/10.1115/1.4042263

Tal H. (2021). C/C++ CSV Writer / Wiki / Home. Retrieved 5th June 2022, from https://sourceforge.net/p/cccsvwriter/wiki/Home/

Tsarau, A., van den Berg, M., Lu, W., Lubbad, R. & Løset, S. (2018). Modelling Results With a New Simulator for Arctic Marine Structures - SAMS. https://doi.org/10.1115/OMAE2018-78593

Tuhkuri, J. & Polojärvi, A. (2018). A review of discrete element simulation of ice–structure interaction [Publisher: Royal Society]. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *376*(2129), 20170335. https://doi.org/10.1098/rsta.2017.0335

Unger, S. (1995). Hazards, critical races, and metastability [Conference Name: IEEE Transactions on Computers]. *IEEE Transactions on Computers*, *44*(6), 754–768. https://doi.org/10.1109/12.391185

van den Berg, M., Lubbad, R. & Løset, S. (2018). An implicit time-stepping scheme and an improved contact model for ice-structure interaction simulations. *Cold Regions Science and Technology*, *155*, 193–213. https://doi.org/10.1016/j.coldregions.2018.07.001

van den Berg, M., Lubbad, R. & Løset, S. (2019). The effect of ice floe shape on the load experienced by vertical-sided structures interacting with a broken ice field. *Marine Structures*, *65*, 229–248. https://doi.org/10.1016/j.marstruc.2019.01.011

Veritas, D. N. (2010). Recommended practice DNV-RP-C205: Environmental conditions and environmental loads. *DNV, Norway.*

Wang, Q. (2014). Design and Dynamic Analysis of a Steel Pontoon-type Semi-submersible Floater Supporting the DTU 10MW Reference Turbine [Publisher: NTNU]. Retrieved 21st February 2022, from https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2400733

White, S., Partlow, J., Sharkey, K., Coulter, D. & Batchelor, D. (2021). Getting Started with Winsock - Win32 apps. Retrieved 14th February 2022, from https://docs.microsoft.com/en-us/windows/win32/winsock/getting-started-with-winsock

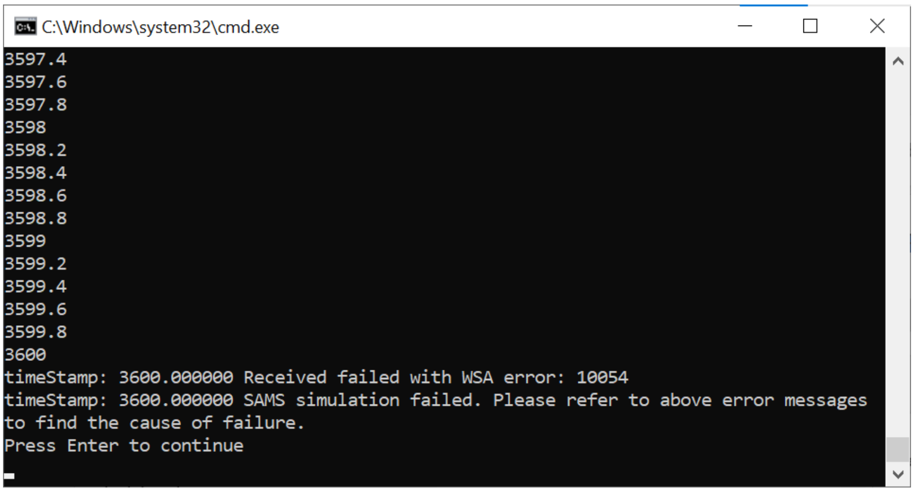
Appendix

Header for coupling code This is the header for the implementation of the coupling algorithm described in Section 3.6. The <winsock2.h> and <ws2tcpip.h> libraries are provided by the Windows operating system. The "cal_conv.h" header is specific to SIMO and provided in Marintek, 2009. The function gfexfo_() is predefined by SIMO and must be declared as it is here. The rest of the functions are all specific to this thesis and are only called from inside gfexfo_().

```
#include <winsock2.h>
#include <ws2tcpip.h>

#include "cal_conv.h"

#pragma comment(lib, "Ws2_32.lib")

#define MSTATE 12
#define MGAMMA 9
#define MCHEXT 120
#define MVELLOT 6

void CAL_CONV gfexfo_(int* iwa, float* rwa, double* dwa, int* ipdms,
        int* iinfo, float* rinfo,
        int* npcur, float* curcor, float* curvel,
        int* kxfo, float* rxfo, float* rhxfo,
        int* ixfo, int* iextf, int* icoord,
        int* nint, int* nrea, int* nsto, int* nstr,
        char chext[][MCHEXT], float state[][MSTATE],
        float gamma[][MGAMMA], float vellot[][MVELLOT],
        float* stor, int* ierr);

SOCKET connect_to_SAMS();

int send_to_SAMS(SOCKET sams_tcp_socket, double time, double
↪ central_forces[6]);

int receive_from_SAMS(SOCKET sams_tcp_socket, double* SAMS_time, double
↪ displacement_SAMS[6], double SIMA_global_to_SAMS_body[3][3]);

int read_from_SAMS(char* SAMS_resultfile_path, int substep_nr_overall, int
↪ nr_of_substeps_overall, double* SAMS_txt_time, double
↪ displacement_SAMS_t0[6], double velocity_SAMS_t0[6], double
↪ F_ice_global[6]);

int get_matrix_from_axis_angle(double e1, double e2, double e3, double
↪ theta, double SAMS_global_to_SAMS_body[3][3]);

int get_joint_rotations_from_matrix(double SAMS_global_to_SAMS_body[3][3],
↪ double joint_rotation[3]);
```

Simulation length [s] input to .itconfig	f [Hz]	Time at end of simulation[s]	Error
18.0	10	18.1	Yes
19.0	10	19.0	No
60.0	10	60.0	No
3600.0	5	3600.2	Yes
3600.0	10	3600.1	Yes
3600.0	20	3600.0	No

Table A.1: Inconsistency with TCP timing in SAMS



```
3597.4
3597.6
3597.8
3598
3598.2
3598.4
3598.6
3598.8
3599
3599.2
3599.4
3599.6
3599.8
3600
timeStamp: 3600.000000 Received failed with WSA error: 10054
timeStamp: 3600.000000 SAMS simulation failed. Please refer to above error messages
to find the cause of failure.
Press Enter to continue
```

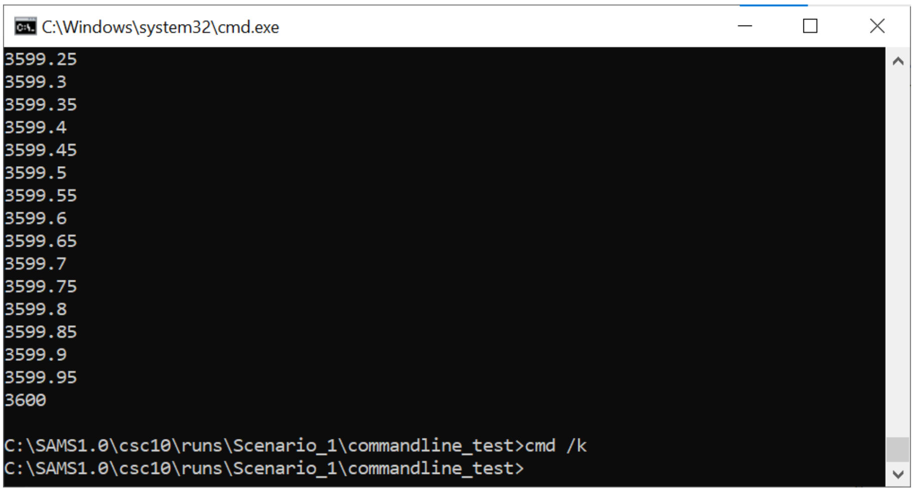
Figure A.1: Example of a 5 Hz simulation ending with a timing error.

A Inconsistency with TCP timing in SAMS

Generally, one message is sent and one message is received over TCP for every timestep. However, there seems to be some inconsistency about whether SAMS performs this TCP communication for the very last timestep. Some information was gathered about this behaviour during the writing of this thesis, collected in Table A.1.

Examples of simulation output with and without errors are provided in Figure A.1 and Figure A.2.

Between these two simulations, no changes were made to any inputs except for the simulation frequency. In both cases, the timestep ending with $t = 3600$ was performed successfully and the SIMA program finished without errors. However, in the first case, SAMS expected to continue simulating, therefore raising an error when the TCP connection closed from the side of SIMA.



```
3599.25
3599.3
3599.35
3599.4
3599.45
3599.5
3599.55
3599.6
3599.65
3599.7
3599.75
3599.8
3599.85
3599.9
3599.95
3600

C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test>cmd /k
C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test>
```

Figure A.2: Example of a 20 Hz simulation ending without a timing error.

B Text output from SAMS

Instead of sending ice forces over TCP, SAMS writes them to a .txt result file some time after finishing the calculating of each timestep. Thus, one must wait until a result row is written by SAMS before proceeding with the coupling.

Although the rows are written after each timestep, the file itself is created at the launch of the simulation, before any connection has been established with SIMA and any calculations have started. At this moment, the file only contains some information about the SAMS build and the column headers that describe the information to be appended in the rows.

Since the time to simulate a timestep in SAMS is unknown, it is also unknown how much time will pass from the moment of sending the TCP input message to SAMS to the moment of attempting to read the file. Since the file is kept open by SAMS all the time, its metadata does not change every timestep. For this reason, the only way to know whether a row has been written is to attempt to read it. The SIMA code, in the process of its execution, might reach this attempt before or after the row has been written. This is known as a race condition (Unger, 1995).

If the row of the current timestep has been already written, then normal execution can proceed. If instead SIMA was faster and the row has not been written yet, the file reading code will reach the end of file. In this case, some time is waited and a new attempt is made. These attempts continue with increasing amounts of waiting time, until SAMS finally writes the row and it can be successfully read. Alternatively, the waiting time can increase to an amount considered unacceptable by the developer. This time duration is called the timeout and if it is reached, the simulation is stopped with an error.

The result file is named according to the simulation name and the time of its start. The name is known, it is taken from the name of the .itconfig file. The timestamp, however, is not known until the simulation has been started in SAMS. One solution would be to start the SAMS simulation, see the according name, and input it to the SIMA GUI before launching the SIMA simulation. This is cumbersome. In this thesis, the code looks for the newest .txt file matching the naming scheme.

After the row has been read from the file, it must be parsed according to the headers of the file. These can vary between SAMS versions. Each entry in the header row contains a name of the value stored in the column, as well as its unit. One exception is the `RotAngle`, which is stored in radians but not declared so in the header.

Additionally, when post-processing the result .txt file, it might be useful to notice that the `RotAxisX` headers are appended without an underscore.

The translational ice forces are given in two separate columns: ice breaking forces and rubble forces. The rotational ice forces are instead given as one column, containing the torques from all types of ice interactions.

C Treatment of timestamps in the information exchange

In the function arguments, TCP packets, and the text files the information is accompanied by a timestamp. There is some variation on the meaning of the timestamp, especially when it comes to forces that are assumed to act from one time instant to another.

Usually, the function argument contains the starting time of the timestep for which the force input is expected. The TCP message from SAMS to SIMA contains the starting time of the timestep for which the force input is expected. The TCP message from SIMA to SAMS contains the starting time of the timestep for which the force input is given. The .txt result row contains the ending time of the timestep for which the forces are recorded.

In some cases when running only SIMO without RIFLEX, it was observed that ending timestamps were passed instead of the starting timestamps. Thus, the time 0.0 would never be passed. This behaviour could not be reliably reproduced, however. At the time of running the simulations for this thesis, RIFLEX was passing the timestamps of the timestep start.

D Information available in SIMA

On every timestep, a function is called with a preset list of arguments described in Marintek, 2009. The ones relevant to the coupling code in this thesis are presented in Table D.1.

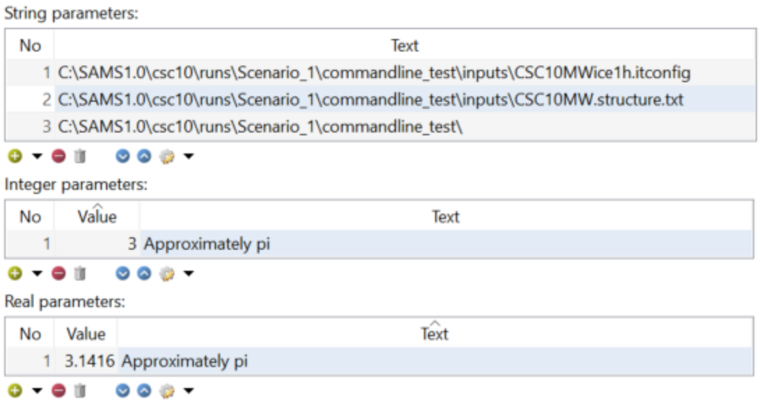
Argument	Input/Output	Type	Description
iinfo[2]	I	int	IBDY, body number
iinfo[5]	I	int	ISTEP, step number
iinfo[6]	I	int	NSTEP, number of steps
iinfo[7]	I	int	IEXTRA, substep number
iinfo[8]	I	int	NEXTRA, number of substeps
iinfo[11]	I	int	ITER, iteration no. in current step
rinfo[0]	I	float	TIME, actual total time
rinfo[1]	I	float	DT, time step length
chext	I	char array	All text input concatenated into one string
state[0]	I	float	Least significant bits of XGLB, the displacement in global X
state[1]	I	float	Most significant bits of XGLB, the displacement in global X
state[2]	I	float	Least significant bits of YGLB
state[3]	I	float	Most significant bits of YGLB
state[4]	I	float	Least significant bits of ZGLB
state[5]	I	float	Most significant bits of ZGLB
state[6]	I	float	Least significant bits of PHI, the last joint rotation
state[7]	I	float	Most significant bits of PHI
state[8]	I	float	Least significant bits of THETA, the second joint rotation
state[9]	I	float	Most significant bits of THETA
state[10]	I	float	Least significant bits of PSI, the first joint rotation
state[11]	I	float	Most significant bits of PSI
stor[0]	O	float	Force in global X to apply to rigid body [kN]
stor[1]	O	float	Force in global Y to apply to rigid body [kN]
stor[2]	O	float	Force in global Z to apply to rigid body [kN]
stor[3]	O	float	Torque around global X to apply to rigid body [MN]
stor[4]	O	float	Torque around global Y to apply to rigid body [MN]
stor[5]	O	float	Torque around global Z to apply to rigid body [MN]

Table D.1: Information passed to the coupling code from SIMA

An output argument marked with "O" means that a memory location was passed to the coupling code, making it possible to write a value there that SIMA would later read.

It should be noted that Table D.1 differs from that presented in Marintek, 2009. The displacement vector, referred in Section 3.4 as $\mathbf{x}_{SIMA_{i-1}}$, is here given through an array called "state". Its twelve values, instead of holding six displacement values and six velocity values, hold half a displacement value each.

In addition to the information described in Marintek, 2009, there are also fields in the user interface for passing arbitrary data. This function is used to provide the coupling code with information about the SAMS simulation, as shown in Figure 15. Some functionality planned in this thesis was originally relying on the ability to pass integer and float parameters as shown in Figure D.1.



String parameters:

No	Text
1	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\inputs\CSC10MWice1h.itconfig
2	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\inputs\CSC10MW.structure.txt
3	C:\SAMS1.0\csc10\runs\Scenario_1\commandline_test\

Integer parameters:

No	Value	Text
1	3	Approximately pi

Real parameters:

No	Value	Text
1	3.1416	Approximately pi

Figure D.1: Input parameters to the coupling code.

In practice, however, only the string parameters were passed successfully from SIMA to the coupling code. All real parameters were changed to 0.0 and all integer parameters were changed to their respective indices in the list.

The coupling code is called on every timestep with a fresh list of function arguments. Some information is also needed from the previous timestep. For this purpose, some variables are declared as static. Since the process that is calling the function is the same, it keeps static variables in memory.

E Global parameters of the simulation in SIMA

In SIMA, there is a difference between the model setup and a condition. Both are stored in .xml format in a SIMA file. However, this file is stored in a different folder than the result folder, which is guaranteed to be accessible by the code.

At the time of running a condition, information about it is written into the results folder. This information can be read by the code, to get information that would otherwise not be available through the function argument.

This code made heavy use of the sys.dat file. This file might be named sys.dat in case of a SIMO simulation or sys-sima.dat in case of RIFLEX simulation. In this file, mass properties of the rigid bodies are given. For the rotational degrees of freedom, the mass matrix coefficients represent a mass moment of inertia. The mass properties are given in units that differ from SI.