



Norwegian University of  
Science and Technology

# Simulation of Vessel Response Time for Emergency Preparedness Against Acute Pollution

**David Josefsen**

Marine Technology

Submission date: June 2016

Supervisor: Bjørn Egil Asbjørnslett, IMT

Co-supervisor: Trond Johnsen, MARINTEK

Norwegian University of Science and Technology  
Department of Marine Technology



# MASTER'S THESIS IN MARINE TECHNOLOGY

SPRING 2016

For stud.techn.

David Josefsen

## Simulation of Vessel Response Time for Emergency Preparedness Against Acute Pollution

### Background

This master's thesis is the result of a joint project between several stakeholders in the Norwegian maritime industry, of which my main collaboration partner is the research institute MARINTEK AS. The project researches the possibilities of conducting an exploration drilling campaign in Arctic areas. Maritime operations in the far north are not only challenging due to harsh weather conditions, but when engaging in Arctic operations, the remoteness of the area can also be troublesome. In such remote areas, one can no longer take advantage of the area emergency preparedness, where cooperation between geographical areas allows different operating companies to share emergency preparedness resources, such as joint standby vessels and SAR helicopters. The demands to emergency preparedness are commonly solved by having a dedicated standby vessel positioned close to the platform, which is tasked with emergency response and monitoring the safety zone around the rig for potential leaks. The drawback is that chartering such a vessel is costly, and it could be economically beneficial to eliminate the need for this extra vessel by establishing a sufficient emergency preparedness system through other means. Thus, we want to investigate what oil spill response time can be achieved through alternative solutions.

### Objective

The objective of this thesis is to develop a simulation model that can evaluate the expected emergency response time of a given fleet composition, and thus serve as a decision support tool for strategic fleet sizing and operational planning.

### Tasks

The candidate is recommended to cover the following parts in the master's thesis:

- a. Review literature relevant to the topic. That means to document what others have done and published previously.
- b. Determine the necessary input data and develop a simulation model that are able to simulate the emergency response time of a given fleet composition.
- c. State limitations and assumptions made in the model, and comment on the validity of the output.
- d. Perform a case study in order to illustrate the application of the developed simulation model.

### General

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

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

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

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

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

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

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

### **Deliverable**

- The thesis shall be submitted in two (2) copies:
- Signed by the candidate
- The text defining the scope included
- In bound volume(s)
- Drawings and/or computer prints that cannot be bound should be organized in a separate folder.
- In case computer programs have been made as part of the thesis work, the source code shall be included. In case of experimental work, the experimental results shall be included in a suitable electronic format.

### **Supervision:**

Main supervisor: Professor Bjørn Egil Asbjørnslett, NTNU

External Supervisor: Trond Johnsen, MARINTEK AS

**Deadline: 10.06.2016**

## Abstract

Maritime operations in the far north are not only challenging due to harsh weather conditions, but when engaging in Arctic operations, the remoteness of the area can also be troublesome. In such remote areas, one can no longer take advantage of the area emergency preparedness, where cooperation between geographical areas allows different operating companies to share emergency preparedness resources, such as joint standby vessels and SAR helicopters. The demands to emergency preparedness are commonly solved by having a dedicated standby vessel positioned close to the platform. The drawback is that chartering such a vessel is costly, and it could be economically beneficial to eliminate the need for this extra vessel. Thus, we want to investigate what oil spill response time can be achieved by utilizing the vessels of the operational fleet for emergency response, in order to remove the supplementary standby vessel. This leads to the objective of this master's thesis:

*The objective of this thesis is to develop a simulation model that can evaluate the expected emergency response time of a given fleet composition, and thus serve as a decision support tool for strategical fleet sizing and operational planning.*

In this thesis we limit the investigated emergencies to oil spills. However, the developed model is also applicable to other emergencies, but may require some minor adjustments depending on the incident investigated. Furthermore, the oil spill is assumed to originate from the offshore installation, and for simplicity, this is chosen as the location of the oil spill. Lastly, it is assumed that the operation takes place in an area where no ice is present.

The simulation model is developed in MATLAB as a discrete-event simulation model, and its function is to determine the fastest feasible response time in case of an emergency. That is, the fastest time one of the vessels of our fleet can arrive at the scene of emergency in order to provide the necessary aid.

The model is tested through a case study, where we seek to determine how the fleet response time changes by the number of vessels carrying oil spill response equipment onboard.

Through the case study, it was shown that the developed simulation model can be utilized to evaluate different fleet compositions with regards to response time, which can help aid in the decision making when determining the fleet specifications with regards to both emergency preparedness and logistics planning.

For an external reader that wish to utilize this model, it is recommended to include some operational limitations given by weather conditions, such as wave height, and also extend upon the relation between weather conditions and their effect on sailing times.

## Sammendrag

Maritime operasjoner i nordområdene er ikke bare utfordrende grunnet tøffe værforhold, men også at Arktiske operasjoner foregår i øde områder kan by på store utfordringer. I slike fjernliggende strøk finnes ikke et etablert områdeberedskap, hvor beredskapsressurser som felles standby fartøy og redningshelikoptre tilgjengeliggjøres for flere operatørselskaper gjennom samarbeid innenfor et geografisk område. Følgelig må det sikres et tilfredsstillende beredskapsapparat for den enkelte operatørs oppdrag i området. Krav til sikkerhetsberedskap og respons er gjerne løst ved å ha et dedikert beredskapsfartøy liggende ute ved installasjonen. Ved områdeberedskap kan ett enkelt skip tilfredsstillende beredskapskrav for flere installasjoner innenfor et område, men i Arktis vil mangel på samarbeidspartnere medføre at denne kostnaden må tas av den enkelte operatør. Følgelig blir det ved Arktiske operasjoner en større kostnadsbesparelse ved å eliminere behovet for et slikt fartøy. På bakgrunn av dette ønsker vi å finne alternative metoder for oljevernberedskap som er tilfredsstillende med hensyn til responstid, og utforsker muligheten for å benytte den operasjonelle flåten som et beredskapsapparat, og følgelig eliminere kostnaden av et ekstra beredskapsfartøy. Dette videreføres inn i masteroppgavens problemstilling, som er formulert som følgende:

*Formålet med denne masteroppgaven er å utvikle en simuleringsmodell som kan vurdere den forventede beredskapsresponstiden for en gitt flåtesammensetning, og dermed fungere som et beslutningsverktøy for strategisk flåtedimensjonering og operasjonell planlegging.*

I denne oppgaven begrenser vi beredskapen til oljevern, og undersøker følgelig oljesøl. Imidlertid er den utviklede modellen også anvendbar for andre ulykkessituasjoner, men kan kreve noen mindre justeringer avhengig av den respektive ulykken. Videre antas offshore installasjonen å være utslippskilden til oljesølet, og benyttes som lokasjonen av oljesølet. I tillegg antas det at operasjonen foregår i et område uten is.

Simuleringsmodellen er utviklet i MATLAB som en diskret-hendelses modell, og dens funksjon er å bestemme den raskest gjennomførbare responstiden for en nødsituasjon. Det vil si, den minste tiden som kreves for at et av flåtens fartøy er tilgjengelig på ulykkesstedet for å bidra med nødvendig hjelp.

Modellens anvendelse testes gjennom et eksempelstudie, hvor vi analyserer endring i flåtens responstid på bakgrunn av antall fartøy i flåten som er installert med oljevernutstyr. Gjennom casestudiet ble det vist at simuleringsmodellen kan benyttes for å evaluere responstid for ulike flåtesammensetninger, og følgelig bistå i beslutningsprosessen vedrørende fastsettelse av flåtens spesifikasjoner med hensyn til både beredskap og operasjonell planlegging.

For en leser som ønsker å benytte modellen i sin helhet eller delvis, så er det anbefalt å inkludere et sett med operasjonelle begrensninger gitt av værforhold, som bølgehøyde, samt å ytterligere utdype relasjonen mellom værforhold og dets påvirkning av seilingstid.



## Preface

This master's thesis has been written by David Josefsen during the spring semester of 2016 at the Norwegian University of Science and Technology, Department of Marine Technology. The thesis amounts for 30 credits, and 100 % of the grade is based on this paper.

During the fall semester of 2015 I wrote a project thesis regarding the application of simulation in the maritime supply chain logistics in arctic region. The project thesis served as a preliminary study for this master's thesis, and provided me with modeling skills for developing simulation models, as well as an introduction to its application in the maritime industry.

The workload has been evenly distributed throughout the semester, with a natural increase in work hours as the deadline approached. The early stages were focused on determining the layout of the simulation model and thesis, as well as doing background research and determining what I wanted to accomplish in the thesis. The intermediate stage was spent on the development of the simulation model, as well as performing the case study analysis and producing results. The bulk of the thesis writing was conducted in the later stages, based on extensive notes and comments written during the two previous phases.

All in all, the thesis work has been both challenging and interesting, and yielded a great learning outcome.

Trondheim, 10.06.2016



---

David Josefsen



## Acknowledgment

Thanks are given to my advisor, Professor Bjørn Egil Asbjørnslett, whom have been available for guidance throughout the entire semester. I would also like to thank Trond Johnsen and Inge Norstad from MARINTEK AS for input on the model development, and Amund Josefsen for feedback and proofreading of the thesis. Lastly, I would like to extend my gratitude to Cecilie Stakkeland for unlimited support throughout the whole work process.

D.J.



## Table of Contents

<b>1 Introduction.....</b>	<b>1</b>
1.1 Background .....	1
1.2 Objective .....	1
1.3 Limitations .....	1
1.4 Current State of Research Regarding Oil Spill Response Time.....	2
1.5 Thesis Structure.....	2
<b>2 Laws and Regulations Valid in the Arctic .....</b>	<b>3</b>
2.1 The Norwegian Oil Spill Response System .....	4
2.1.1 Barriers .....	5
<b>3 Simulation Approach to Oil Spill Preparedness .....</b>	<b>7</b>
3.1 Applied Methodology .....	7
3.2 Review of Previous Work and Findings in the Literature.....	9
<b>4 Model Applied for Simulation of Oil Spill Response Time .....</b>	<b>13</b>
4.1 Input Data Required in The Simulation .....	15
4.1.1 Improving Model Run Time by Forcing Excel to Remain Open .....	15
4.1.2 Input Parameters .....	16
4.1.3 Fleet Characteristics .....	17
4.1.4 Fleet Status .....	18
4.1.5 Locations of Interest .....	20
4.1.6 Metocean Data .....	20
4.2 Mapping and distance calculations .....	24
4.3 Calculating Sailing time .....	26
4.4 Estimating the Total Fuel Consumption for the Response.....	27
4.5 Determining Feasibility with Regards to Current Fuel Level .....	28
4.5.1 Direct Response by an Individual Vessel .....	28
4.5.2 Direct Response Requiring Refueling Offshore .....	29
4.5.3 Refueling in Port.....	30
4.6 Determining a Feasible Response .....	31
<b>5 Case Study: Response Times for a Varying Number of NOFO-Equipped Vessels .....</b>	<b>33</b>
5.1 Stochastic Input Data .....	35
5.1.1 Current Fuel Level.....	36
5.1.2 Operational Status.....	36

5.1.3 Cargo Status.....	37
5.1.4 Location.....	37
5.1.5 Metocean Data.....	39
5.2 Results of the Case Study for a Varying Number of NOFO-Equipped Vessels.....	40
5.2.1 Three Vessels Installed with NOFO-Equipment.....	41
5.2.2 Two Vessels Installed with NOFO-Equipment.....	43
5.2.3 One Vessel Installed with NOFO-Equipment.....	45
5.2.4 No Vessels Installed with NOFO-Equipment.....	47
5.3 Conclusion of the Case Study.....	50
<b>6 Discussion.....</b>	<b>51</b>
<b>7 Conclusion and Further Work.....</b>	<b>53</b>
7.1 Conclusion.....	53
7.2 Further Work.....	54
<b>Reference List.....</b>	<b>55</b>
<b>Appendix.....</b>	
Appendix A: The Simulation Model (MATLAB Script).....	A
A.1 Initialization.....	A
A.2 Input Data.....	B
A.3 Calculations.....	A
A.4 Determining the Response.....	A
A.5 Plot the Solution.....	A
Appendix B: xlsread1.m.....	B
Appendix C: Input Data Sheet.....	C
Appendix D: Case Study Simulation Model.....	D
Appendix E: Abbreviations.....	E

## Tables

Table 1: Simulation Input Data.....	16
Table 2: Speed modes for each vessel .....	17
Table 3: Fuel consumption modes .....	17
Table 4: Operational Status Variable .....	18
Table 5: Cargo Status Variable .....	19
Table 6: Emergency Equipment Availability Variable.....	19
Table 7: Locations of Interest .....	20
Table 8: Metocean Data Set Template.....	20
Table 9: Metocean Data Parameters .....	21
Table 10: State-Transition Matrix for Weather Conditions.....	22
Table 11: Distances of the great circle routes .....	25
Table 12: Speed modes for each vessel .....	26
Table 13: Fuel consumption modes .....	27
Table 14: Response Options with Regards to Fuel.....	28
Table 15: The Four Cases Simulated .....	34
Table 16: Stochastic Input Data.....	35
Table 17: Choosing the Operational Status Variable.....	36
Table 18: Maximum Response Time Within a 1 % and 5 % Significance Level .....	48
Table 19: Mean Response Time for the Four Cases Simulated.....	49

## Figures

Figure 1: Principle Illustration of the Multi-Barrier System (NOFO,2016).....	6
Figure 2: Overview of the Simulation Model.....	14
Figure 3: Layout of Input Data in Excel .....	15
Figure 4: Relations and Purpose of the Input Data .....	16
Figure 5: Mercator Projection Showing the Distortion of Areas Further Away from the Equator (Støwer,2015).....	24
Figure 6: Decision Tree – Deciding a Feasible Response Action for Each Vessel .....	31
Figure 7: Plotted Solution Example.....	32
Figure 8: The Sailing Route Divided into Three Rectangular Areas Where the Vessels Are Assumed to Operate.....	37
Figure 9: The Sailing Route Defined by Only One Rectangular Area .....	38
Figure 10: Density Plot of the Response Times - Three Vessels Equipped .....	41
Figure 11: Probability Plot of the Response Times - Three Vessels Equipped (0.01 Significance Level) .....	42
Figure 12: Density Plot of the Response Times - Two Vessels Equipped .....	43
Figure 13: Probability Plot of the Response Times - Two Vessels Equipped (0.01 Significance Level) .....	44
Figure 14: Density Plot of the Response Times - One Vessel Equipped.....	45
Figure 15: Probability Plot of the Response Times - One Vessel Equipped (0.01 Significance Level) .....	46
Figure 16: Density Plot of the Response Times - No Vessels Equipped.....	47
Figure 17: Probability Plot of the Response Times - No Vessels Equipped (0.01 Significance Level).....	48





## Chapter 1

### Introduction

#### 1.1 Background

This master's thesis is the result of a joint project between several stakeholders in the Norwegian maritime industry, of which my main collaboration partner is the research institute MARINTEK AS. The project researches the possibilities of conducting an exploration drilling campaign in Arctic areas. Maritime operations in the far north are not only challenging due to harsh weather conditions, but when engaging in Arctic operations, the remoteness of the area can also be troublesome. In such remote areas, one can no longer take advantage of the area emergency preparedness, where cooperation between geographical areas allows different operating companies to share emergency preparedness resources, such as joint standby vessels and SAR helicopters (PSA, 2005). The demands to emergency preparedness are commonly solved by having a dedicated standby vessel positioned close to the platform, which is tasked with emergency response and monitoring the safety zone around the rig for potential leaks. The drawback is that chartering such a vessel is costly, and it could be economically beneficial to eliminate the need for this extra vessel by establishing a sufficient emergency preparedness system through other means. Thus, we want to investigate what oil spill response time can be achieved through alternative solutions. More precisely, by utilizing the vessels of the operational fleet for emergency response in order to remove the need for a supplementary standby vessel. This leads to the objective of this master's thesis.

#### 1.2 Objective

*The objective of this thesis is to develop a simulation model that can evaluate the expected emergency response time of a given fleet composition, and thus serve as a decision support tool for strategical fleet sizing and operational planning.*

#### 1.3 Limitations

In this thesis we limit the investigated *emergencies* to *oil spills*, and will use these two terms synonymously throughout the paper. However, the developed model is also applicable to other emergencies, but may require some minor adjustments depending on the incident investigated. Furthermore, the oil spill is assumed to originate from the offshore installation, and for simplicity, this is chosen as the location of the oil spill. In reality, the oil spill may drift in some direction before the response is on site. However, this could result in both

longer or shorter response times, depending on the currents. Thus, the point of origin should make for an acceptable intermediate position. Lastly, it is assumed that the operation takes place in an area where no ice is present.

#### 1.4 Current State of Research Regarding Oil Spill Response Time

As mentioned, common practice is to have a dedicated response vessel in standby close to the platform, which is available for response at short notice. Thus, the time it takes for a response vessel to be available on site has not been a very relevant topic for study. Consequently, available literature found has been regarding simulation of the actual oil recovery operations as well as the expected drifting and weathering of the oil spill, and papers regarding the expected response time of the vessels.

#### 1.5 Thesis Structure

Regarding the structure, the thesis will first provide the reader with some insight on the Norwegian oil spill response system, and some of the regulations one are subject to. Furthermore, the thesis will briefly present some background information on simulation methodology, and comment on current literature within this field of study. We then dive into the more technical details of this paper, where the developed simulation model is presented in full extent, from input to output. After presenting the simulation model, a case study is conducted in order to illustrate its application through a relevant problem example. In the case study, we analyze how the response time can be expected to change depending on the number of vessels installed with emergency equipment onboard. This serves as an important input when deciding how many vessels are necessary to equip with oil spill response equipment in order to satisfy any safety demands. The basis of this problem is that oil response equipment consumes deck area. Consequently, we wish to equip as few vessels as possible, while maintaining an adequate safety function, so that as much deck area as possible can be utilized for cargo transport. Finally, we discuss the application and validity of the developed simulation model, and further developments that could be looked into at a later stage.

## Chapter 2

### Laws and Regulations Valid in the Arctic

Arctic areas, such as the Barents Sea, is characterized by lack of infrastructure and harsh weather conditions with low temperatures. This is challenging in several aspects, including emergency preparedness. However, there are no special regulations for petroleum activity in the Arctic, and the ordinary Norwegian HSE-regulations are as valid here as for any other part of Norwegian waters. In Norway, the safety regulations are formulated for functionality, which means that the regulations do not directly specify any solutions, but rather form demands to what level of safety must be achieved. In practice, this means that the demanding conditions of the Arctic may call for different technical solutions compared to operations further south. It is, however, up to the petroleum companies themselves to account for any challenges specific for their operation.

We have established that the ordinary regulations are applicable for Arctic operations on the Norwegian continental Shelf. Consequently, the development of a sufficient emergency preparedness system is subject to the several regulations given by Norwegian Law and the Petroleum Safety Authority Norway (PSA), whereas the main regulations one should be aware of are listed below:

- Act Pertaining to Petroleum Activities:
  - § 9-2. Preparedness
- The Management Regulations:
  - § 17 Risk Analyses and Emergency Preparedness Assessments
- The Activities Regulations:
  - § 73 Establishment of Emergency Preparedness
  - § 76 Emergency Preparedness Plans
  - § 79 Action Against Acute Pollution
- The Framework Regulations:
  - § 11 Risk Reduction Principles
  - § 20 Coordination of Offshore Emergency Preparedness
  - §21: Offshore Emergency Preparedness Cooperation

### 2.1 The Norwegian Oil Spill Response System

The national emergency preparedness system combines both public and private oil spill response resources, following standards set by the Norwegian Clean Seas Association for Operation Companies (NOFO). The cooperation between public and private response organizations ensures that the entire national emergency response apparatus on the Norwegian continental shelf are available at all times. Each organizations responsibility regarding oil spill response, including clean-up and restoration, is clearly defined and established in the following three regimes (Klaussen, 2013):

- In case of oil spills from ships, the Norwegian Coastal Administration (NCA) is responsible for oil spill response. If deemed necessary, NCA may order mobilization of all national oil spill response resources.
- In case of oil spills from offshore installations, the operator is responsible for oil spill response. This is generally executed using response resources provided by NOFO. As for the NCA, NOFO may call upon all national oil spill response resources to aid in the emergency, if necessary.
- In case of minor and local incidents, the respective municipal emergency response organization is responsible for response.

In this master's thesis, we assume that the oil spill originates from the offshore installation, making the operator responsible for the emergency response.

### 2.1.1 Barriers

The primary strategy against acute oil pollution on the Norwegian Continental Shelf utilizes mechanical recovery systems in areas close to the spill source. Additionally, chemical dispersion is also a feasible option that should be considered. According to the Pollution regulations §19 – 6, chemical dispersion should be used when more beneficial to the environment than mechanical recovery methods.

No emergency preparedness measures are in themselves perfect. Therefore, The Norwegian oil spill preparedness system is based on a multi-barrier principle, implementing several barriers, both as preventive measures and in order to reduce the impact if an accident occurs. The preparedness system is divided into the five following barriers, illustrated in figure 1:

- Barrier 0: Safety barriers that may prevent the oil spill from the offshore installation
- Barrier 1: Recovery near the source of the oil spill
- Barrier 2: Recovery of the drifting oil spill in open sea
- Barrier 3: Recovery in coastal waters
- Barrier 4: Coastland and stranded oil cleaning

The barrier system must be established in accordance to the Management Regulations § 5 given by the Petroleum Safety Authority Norway.

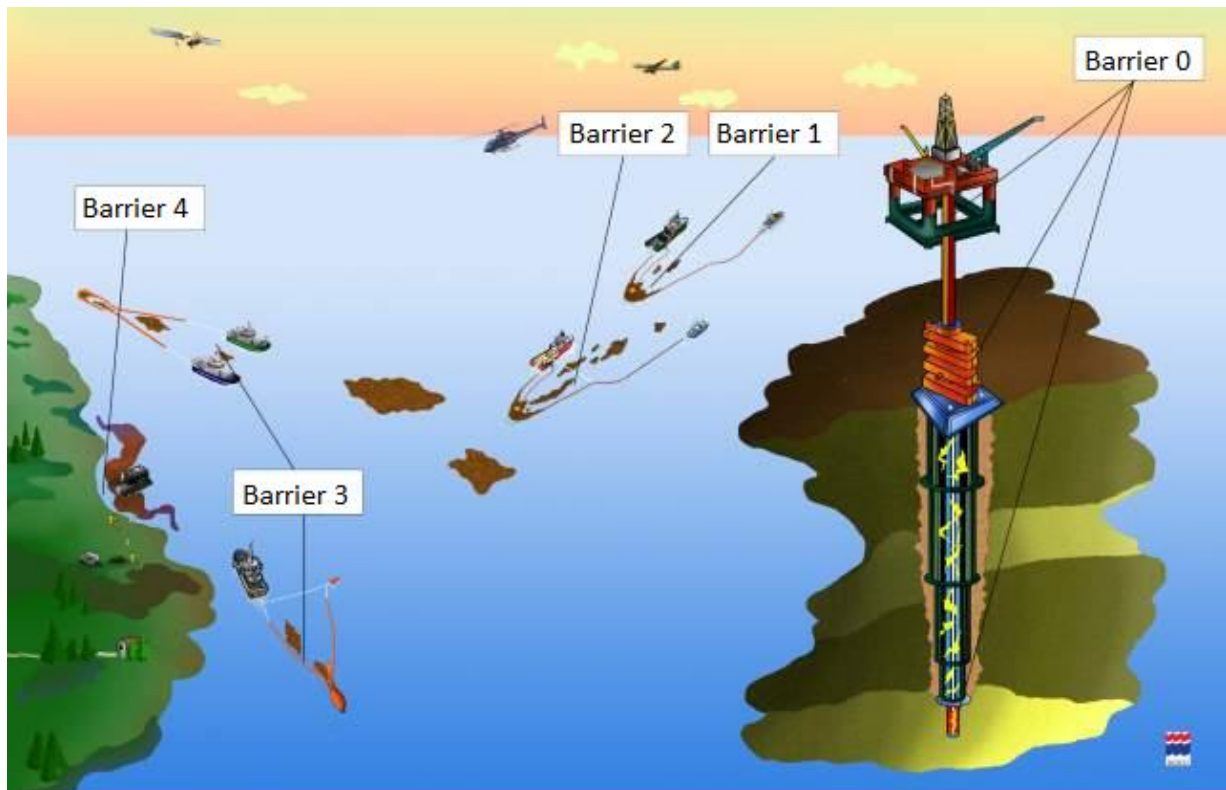


Figure 1: Principle Illustration of the Multi-Barrier System (NOFO,2016)

In this thesis, we look into the establishment of the first barrier. With regards to HSE, the oil spill is allowed to drift for an hour or two, as to create some distance to the installation before initiating the recovery operations. This is done to avoid any further accidents due to oil evaporation, fire or explosions. Additionally, this allows for the oil viscosity to increase, making the retrieval process easier (NOFO, 2013).

In terms of performance, the Norwegian Oil and Gas Association guidelines states the following minimum system requirements of the first two barriers (Norwegian Oil and Gas Association, 2013):

*“Barriers at open sea (1 and 2) shall each have sufficient capacity to handle the available emulsion based on the dimensioning rate. Furthermore, the response time for a fully developed barrier shall be shorter than the 5-percentile of the stranding time.”*

## Chapter 3

### Simulation Approach to Oil Spill Preparedness

#### 3.1 Applied Methodology

This chapter aims to provide an introduction to simulation methodology, the benefits of using it, and the major drawbacks.

In the real world, most complex systems include stochastic elements, and are difficult to accurately describe by mathematical models that can be analytically evaluated (Aneichyk, 2009). Thus, we state a set of assumptions in order to design mathematical or logical models that provides a sufficient representation of a real system, and run simulations with that model in order to predict the behavior of said system. Consequently, simulation enables us to analyze the current or future performance of a planned or existing system, instead of executing and observing a real life one.

Developing a simulation model is a fine art of balancing detail and complexity. It is important to design a model in such depth that the output is representative of the behavior of the real system. However, we want to keep the model as simple as possible, as complexity may cause the results to be misinterpreted, and it is easier to both make and overlook potential errors. Additionally, there is often a limited time frame when developing simulation models for real projects, which further supports the idea of making models as simple as possible without compromising the validity of the output. Thus, our goal is not to design a model that is 100 % true to the real process, but a model that provides a useful representation and results that resemble the reality.

As mentioned, when we seek to imitate real world operations, it is necessary to establish a set of assumptions describing the behavior of the real world system. These assumptions are represented as mathematical and logical relationships, which describes the real life processes. When simple enough, these relationships may be analyzed by means of mathematical methods. However, most real world systems are complex and include stochastic elements, and must therefore be evaluated by use of simulation. This is usually performed by an appropriate computer software. Today, there exists several such software programs, but regular script languages such as MATLAB and Java may also be applied to develop a simulation model.

The major drawbacks of simulation, which may prevent simulation from being successfully implemented in certain projects, are model-development time and the modelling skills required of the user (Carson II, 2005). With regards to model development time, the limited time-frame present in many projects may cause the simulation to be insufficiently developed, and the study may produce unreliable and misleading results. Regarding the second impediment, inexperienced analysts may also commit too much time in the model development and detail work, which may not improve the validity of the results as the model is mainly based on uncertain parameters. Additionally, the necessary input data needed for the simulation can often be both costly and difficult to obtain.

The type of simulation utilized in this thesis is discrete-event simulation, which means that the system operation is modeled as a discrete sequence of events in time; a change in the system state is triggered by the occurrence of an event at a discrete point of time. What happens in between successive events is thus not relevant, as no change is assumed to occur in the system state within that time frame, consequently saving a lot of the computational resources demanded.



### 3.2 Review of Previous Work and Findings in the Literature

This chapter will present some contributions relevant to the topic of emergency preparedness against acute oil spills on the Norwegian continental shelf, with a focus on developments in Arctic regions. We review both literature that looks into the general aspect of emergency preparedness and oil spill response, as well as simulation models developed for both oil spill response and some logistical planning problems.

Norconsult AS (2010) published a report regarding preparedness against acute oil pollution in the Northern and Arctic areas. The paper presents the Norwegian oil spill response system and its multi-barrier principle for oil spill recovery. Furthermore, the paper evaluates the progress and development of the Norwegian emergency preparedness system over the last few years, addresses the current status of the national response resources, and the expected development and challenges for the years to come. Norconsult concludes that a large number of advances collectively have led to a substantial improvement of preparedness against acute pollution in the recent years. Additionally, Norconsult expects a significantly higher efficiency of the oil spill preparedness in the Arctic in the coming years.

The Norwegian Oil and Gas Association (2013) presents a guidance for environmental contingency analyzes, a document developed by a joint network of professionals within the field of environmental risk and oil spill response. The purpose of the paper is to describe the framework and implementation of the environmental emergency preparedness analysis, and to provide guidelines to ensure unambiguous input data, assumptions and principles used in the analysis. The paper does not recommend a specific analysis methodology, but provides general guidelines intended to ensure that the preparedness against acute pollution is analyzed and dimensioned in a unified and responsible manner.

When developing a simulation model for operations in the Arctic, there are certain elements that may need some extra attention based on the given project, such as area remoteness. Nordbø (2013) investigates the challenges related to area remoteness, with the objective to improve the supply service for remote locations offshore. Based on a case study, Nordbø proposes to use two converted bulk ships as storage units located at the oil field, where one ship stacks up on supplies, while the other supplies the vessels operating at the oil and gas field. This will reduce the cost of the supply service for remote location, given that the supply demand is large enough, and the location is sufficiently far away from shore.

Ulstein (2014) studies the application of simulation for Arctic field logistics. Through two case studies, Ulstein investigates whether a simulation model can be used to determine the operational duration and optimal fleet composition of platform supply vessels operating in the Arctic. The first case study confirms that the simulation model is capable of analyzing environmental impact on the platform supply vessels' operational duration, and the results from the second case study shows that also an optimal fleet composition can be found. The optimal fleet composition is determined by combining the discrete-event simulation model with a genetic algorithm for optimization. Ulstein puts emphasis on the environmental conditions present in the Arctic, and provides a set of operational limitations that is implemented in the simulation model, which includes wave height, wind speed, visibility, temperature, polar lows and drifting ice. Polar lows and drifting ice forces operation shutdown if present within a given radius, while the other limitations activates if the respective values exceeds a set threshold. If exceeded, the vessels are modeled to wait for appropriate weather conditions, which is updated at given time intervals, e.g. every three hours for wind and wave height.

Victor Westerberg (2012) develops a simulation model for decision aid in response operations that determines possible countermeasures and the extent of an Arctic oil spill response force for a given oil spill. The focus of Westerberg's thesis is to evaluate the available and feasible response methods, and determine the most effective countermeasure against the oil spill. It does not focus on the logistical part of this operation, but rather on the response techniques and methods, such as mechanical recovery, chemical dispersants and in-situ burning.

SINTEF Materials and Chemistry (2014) have developed a simulation model called OSCAR (Oil Spill Contingency and Response), which is a tool for planning and response for oil spills. SINTEF states that the model can be applied for oil spill risk evaluation, oil spill response planning and modeling of oil spill response operations. OSCAR is utilized for predicting the effects and behavior of oil released during an accident, as to evaluate the effects of contingency and response methods. As in Westerberg (2012), this model focuses more on the response measures themselves, and the behavior of the oil during an accident when subject to weathering and currents.

Regarding state-of-the-art simulation models that includes stochastic elements for maritime fleet sizing and planning problems, both stochastic weather conditions, sailing times, and supply demand have been successfully implemented. Aneichyk (2009) develops a discrete-event simulation model for the offshore supply process that can serve as a tool for strategical fleet sizing and operational planning. The model implemented stochastic weather conditions, delays of supplies on the onshore base, and allowed calls for extra visits on demand by offshore installations. These stochastic factors were shown to influence the weekly vessel plan, which could result in delays and a shortage of supply vessels. Aneichyk suggests to hire vessels from the spot market as to satisfy the supply demand, although spot-charters would normally be more expensive than vessels on long-term contracts. The uncertainties make it difficult to apply analytical methods, and simulation is thus more suitable in this case.

Maisiuk and Gribkovskaia (2014) address the planning problem for service of oil and gas offshore installations by a fleet of supply vessels, with stochastic sailing and service times. The discrete-event simulation model is used in the evaluation of alternative fleet compositions, while taking the stochastic weather conditions, such as significant wave height and wind speed, and future spot vessel rates into consideration. The authors state that determining the fleet composition a year ahead has a significant economic effect on the annual vessel costs.

It was experienced that several scientific papers related to the subject had either restricted access or required purchase. Moreover, most literature found focused on the simulation of weathering effects and drifting of the oil spill, as to determine the most effective oil response and recovery actions based on a continuous update of the oils characteristics. Additionally, several papers provided a general evaluation of the Norwegian emergency preparedness system against acute pollution as a whole, in addition to evaluation of various spill containment methods and their usage.

That the papers focus on the recovery operations could be expected, as it is common practice to have a dedicated standby vessel in close proximity to the drilling operations, which is readily available at the accident area on short notice. Consequently, there has not been a need for a dedicated study of the logistical aspect concerning the availability of a response vessel, and the initial response time required for the vessel to be on site. As we have found no current literature addressing this topic, we can conclude that this master's thesis contributes to pioneering research on the subject.



## Chapter 4

### Model Applied for Simulation of Oil Spill Response Time

The simulation model is developed in MATLAB. There also exists an extension to MATLAB called SimEvents, which provides a more interactive graphical modeling environment for discrete-event simulation. However, as SimEvents utilizes a graphical drag-and-drop approach, it was decided to rather write out the entire model in MATLAB code. This is because the graphical building blocks in SimEvents contains modeling functions that are not shown by code, which makes it difficult to see exactly how the model processes data. By putting the entire model in code also makes it more accessible for other individuals across different programming platforms to study the model in detail, in order to make improvements or find inspiration for their own developments.

The function of the model is to determine the fastest feasible response time in case of an emergency. That is, the fastest time one of the vessels of our fleet can arrive at the scene of the emergency in order to provide the necessary aid.

The simulation model is developed as an extension to MARINTEK's model, which focuses on the logistical aspect of the daily operation. The two models are intended to work in unison, where the logistic model simulates the daily operation, and in the event of an oil spill, provides all necessary information regarding the current status of the fleet and oil spill, in order for this thesis' model to determine the course of action and the expected time for the first response to be operative.

Based on given input data, the simulation model performs the necessary calculations, and runs through a series of logical decisions in order to determine the lowest feasible response time of a vessel. A flow chart illustrating the structure and general outline of the model is presented in the figure 2.

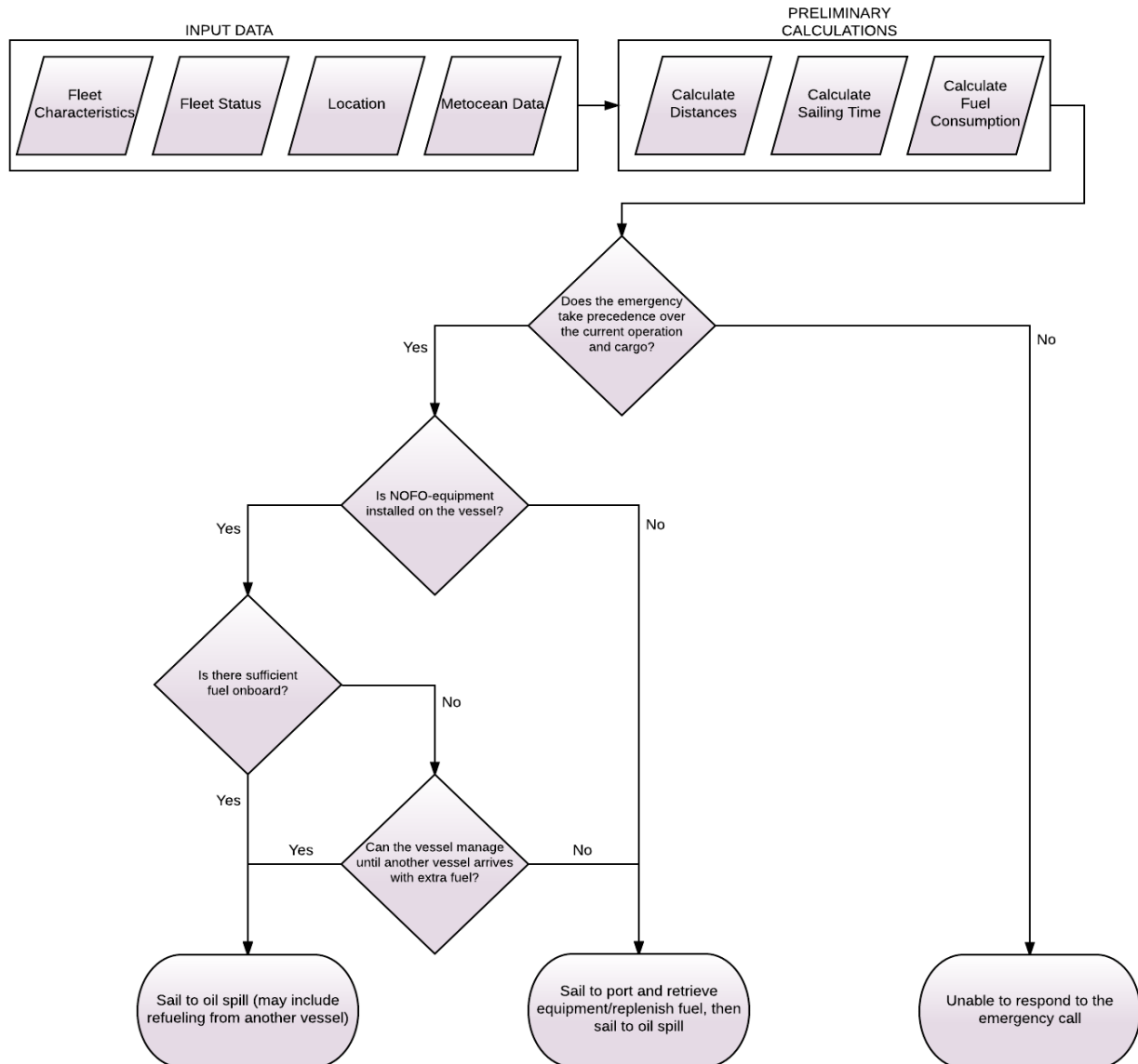


Figure 2: Overview of the Simulation Model

This illustrated process is repeated for every vessel of the fleet, and the minimum feasible response time is determined for each individual ship. Subsequently, the vessel that provides the lowest feasible response time of the entire fleet, is tasked with first response to the emergency. The solution is further evaluated in order to determine the specifics of the response plan.

This chapter will present each step of the model in further detail, starting with how the input data is defined, then going through the calculations performed by the model, and lastly, how a solution is chosen.

#### 4.1 Input Data Required in The Simulation

The simulation input is read from an accompanying Excel-file, where each parameter is defined column by column, as illustrated in figure 3. The Excel-sheet containing all input data can be seen as a whole in *Appendix C: Input Data Sheet*, and the code specifics for how the data is read can be further studied in *Appendix A.1: Input Data*.

Vessel nr	Max Speed [knots]	Service Speed [knots]	Economic Speed [knots]
1	14.6	12	10
2	14.6	12	10
3	14.6	12	10

Figure 3: Layout of Input Data in Excel

The model will initialize by reading down the column containing the number of vessels until the occurrence of an empty cell, and by doing so, determining the first and last row of the remaining columns of the data set so that the appropriate data is all included.

##### 4.1.1 Improving Model Run Time by Forcing Excel to Remain Open

When reading from Excel in MATLAB, the default function has the flaw that it opens and closes Excel every time the function is called. This means that the process *open Excel – read data – close Excel* is repeated for every input parameter in the model, causing the model run time to be long. To combat this, we make use of the function *xlsread1.m* (Antonio, 2008). This function was created and published by user Antonio of the Federal University at Campina Grande on the MathWorks official forums in December 2008, and is attached in *Appendix B: xlsread1.m*. A few lines of code are added in the beginning and end of our program, which respectively works to open and close the ActiveX server, meaning that the Excel-file containing the input data can stay open throughout the whole run of the program. We replace the MATLAB built-in function *xlsread.m* with the new function *xlsread1.m*, and now the program needs only to open and close Excel once while reading the various input parameters. This improvement drastically reduced the run time of the program, and was crucial for the performance when running several simulations for the case study in *chapter 5*. A quick timed test run revealed that even with as few as fifteen simulations, the run time was reduced by almost 80 % of the original run time by implementing *xlsread1.m*.

4.1.2 Input Parameters

The input data can be divided into the groups as seen in table 1, according to the nature of the input parameter. The relations of the various input parameters and their use in the model, is illustrated in figure 4.

Table 1: Simulation Input Data

Fleet Characteristics	Fleet Status	Location	Metocean Data
Number of Vessels	Current Fuel Level	Vessel	Significant Wave Height
Speed	Operational Status	Port	Date
Fuel Consumption	Cargo Status	Oil Spill	
Fuel Capacity	Emergency Equipment	(Refueling rate)	

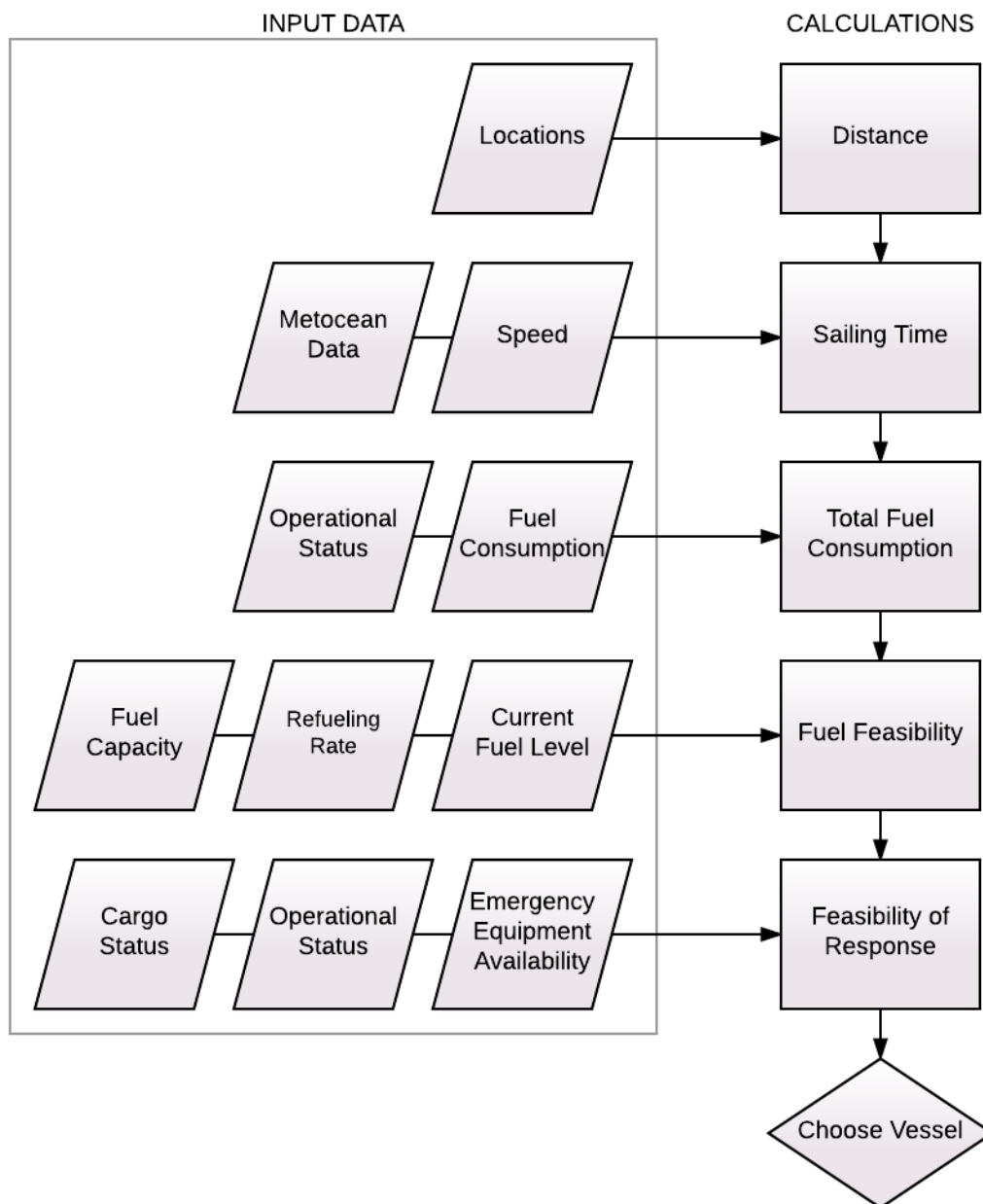


Figure 4: Relations and Purpose of the Input Data



The following sections of this chapter will go more in depth of each parameter and how it is defined for use in the simulation model.

#### 4.1.3 Fleet Characteristics

First and foremost, the fleet size is determined, and further characteristics and details are defined for each individual vessel, beginning with the vessel speed.

##### 4.1.3.1 Speed

The vessel speed is given in [knots] and defined for three separate operation modes for each vessel, as shown in table 2. Depending on the operation at hand, the appropriate speed mode is used in the calculation. As an example, when responding to an emergency, we will base our calculations on the maximum speed, as we seek to minimize the response time.

Table 2: Speed modes for each vessel

Speed Modes
Maximum speed
Service speed
Economic speed

##### 4.1.3.2 Fuel Consumption

As the power output varies for each speed mode, so does the fuel consumption. In addition to defining the fuel consumption for each speed mode, it is also necessary to include the fuel consumption of the vessel when performing oil spill recovery operations, which is needed to determine the feasibility of a vessel response in *chapter 4.5 Determining Feasibility with Regard to Current Fuel Level*. The fuel consumption modes are shown in table 3, and are given in [tonne/day].

Table 3: Fuel consumption modes

Fuel consumption modes
Fuel consumption in maximum speed
Fuel consumption in service speed
Fuel consumption in economic speed
Fuel consumption in operation

##### 4.1.3.3 Fuel Capacity

The total fuel capacity of each vessel is to be given in [tonne], and is necessary in order to estimate the bunkering time of the vessel.

#### 4.1.4 Fleet Status

In addition to the technical specifics of the fleet, we also require information about the current state of each vessel when the incident occurs.

##### 4.1.4.1 Current Fuel Level

The current fuel level of the vessel should be given in [tonne], and is needed in order to evaluate whether there is sufficient fuel to respond to- and combat the oil spill, as discussed in *chapter 4.5 Determining Feasibility with Regard to Current Fuel Level*.

##### 4.1.4.2 Operational Status

The operational status tells us whether the vessel is part of an ongoing operation that it cannot abort, or not. If it is not, it should be further specified the general direction in which the vessel is headed, so that we may include delays caused by e.g. turnaround time, or navigation in/out of port. This is included in the model by assigning an integer value to the variable, where each value is representative of a state of operation as described in table 4.

Table 4: Operational Status Variable

Variable Value	Meaning
0	Unable to abort current operation
1	In port
2	In transit, sailing towards the platform (or at the platform)
3	In transit, returning to port

It should be noted that if a vessel is not eligible for combatting the oil spill, because it cannot abort its current operation, it is simply deemed as a non-feasible solution, and taken out of the evaluation. It is not checked for how long it is tied up with its current operation, and could in theory finish the operation and still be ready for tackling the oil spill before any other vessel arrives. However, estimating how long the operation should take to finish would require an extensive source of data regarding every operation the fleet commences, which would be both rather unique and vary greatly from task to task. Furthermore, it is assumed that for most cases the oil spill takes precedence over all, and the occurrence of an operation that cannot be halted in favor of battling the oil spill is rare. So, when running several simulations, the impact of this simplification should not be significant, and the extra work and computational resources required to include details on every vessel's operation is not justifiable.

#### 4.1.4.3 Cargo Status

The variable *Cargo Status* is a binary variable, that tells us whether the current cargo of the vessel will interfere with its ability to respond to the emergency call, as seen in table 5.

Table 5: Cargo Status Variable

Variable Value	Meaning
0	The cargo causes conflict with the emergency response
1	The cargo does not conflict with emergency response

As for the Operational Status variable, if a vessel is carrying a cargo whose delivery is prioritized over responding to the oil spill emergency (i.e. the variable value is 0), it is not further investigated when the vessel will be available, and the vessel is simply deemed unfit for emergency response. However, this is assumed to not occur frequently and should thus not affect the validity of the results to any significant extent.

#### 4.1.4.4 Emergency Equipment Availability

The availability of emergency equipment onboard the vessel is also given as a binary variable, as seen in table 6

Table 6: Emergency Equipment Availability Variable

Variable Value	Meaning
0	Emergency equipment is not available
1	Emergency equipment is available

Unlike the two previous status variables, the lack of emergency equipment onboard does not eliminate the vessel as a solution for emergency response, but rather forces the vessel to first travel to port and pick up the necessary equipment before setting out to the oil spill site.

4.1.5 Locations of Interest

The location of each point of interest, as well as the whereabouts of the vessels, are given in latitudes and longitudes. Table 7 below shows all locations that are essential in the model.

Table 7: Locations of Interest

Locations of Interest
Vessels
Oil Spill / Offshore Installation
Port

Additionally, the refueling rate of which the port is able to supply a vessel with fuel should be given in [tonne/hour].

4.1.6 Metocean Data

The last bulk of simulation input is the metocean data. This data is given in a separate csv-file (comma separated values). In this model we use a weather data set from waveclimate.com (Ocean Systems Simulation, 2015) which contains buoy-readings for every three hour intervals in the period 1992 – 2012. The data set follows a template as illustrated in table 8.

Table 8: Metocean Data Set Template

Parameters:	Year	Month	Day	Hour	u10	u10d	Hs	Hsd	Tz	Tm	Tp
Unit:	-	-	-	-	[m/s]	[deg]	[m]	[deg]	[s]	[s]	[s]
Example:	1992	01	01	00	10.80	118.0	1.12	117.0	3.82	3.97	3.56
	1992	01	01	03	13.00	108.0	1.49	109.0	4.16	4.38	3.91
	...										
	2012	12	31	21	18.40	136.0	6.22	165.0	8.46	9.92	10.15

With parameters as defined in table 9.

Table 9: Metocean Data Parameters

Parameter	Definition
u10	One-hour wind speed at 10 meters above sea level
u10d	Wind direction at 10 meters above sea level (nautical)
Hs	Significant wave height
Hsd	Mean wave direction (nautical)
Tz	Zero-crossing wave period
Tm	Mean wave period
Tp	Peak wave period

Due to seasonal variations, one should make sure to use weather data that are representative for the time of the year the operation aims to represent. In other words, one should try to avoid using weather readings from July to simulate operations in February. In this model we solve this by defining the current month and day as input. As seen in the column titled *Month* in table 8, the months are defined numerically from 1 – 12, beginning with January. For simplicity, the same format is chosen for how the month should be defined in the input data. Also, the days are to be given numerically as 1 – 31 and so on, depending on the number of days in the respective month.

And so, knowing the month and day for which we seek to simulate operations, the model will read all historic data in the weather data set for that same month and day. E.g. if the current date is the 20<sup>th</sup> of March, then the model will read the weather data for every 20<sup>th</sup> of March in the time period 1992 – 2012, adding up to a total of 168 readings per parameter for that specific date. Finally, one of the many 20<sup>th</sup>-of-March-readings is chosen at random to represent the weather conditions for this particular simulation run.

As the data set contains all buoy-experienced weather within a 21-year span, and the model can in theory select any of these buoy-readings, this method should naturally provide a good range of weather options that are both accurately distributed and statistically valid. If we assume the weather to behave similarly to what was measured between 1992 and 2012, a fair assumption, then this data set already contains weather from one extreme to the other, and all in between. Furthermore, the frequency of certain weather will be realistically represented in the model, as the statistical chance of choosing a reading of harsh weather would be similar to the probability of harsh weather to actually occur for that geographical area.

#### 4.1.6.1 Sampling Metocean Data for Short Sailing Distances

For this thesis, the geographical area in which the vessels operate is of a smaller scale, where the distance between port and offshore installation is relatively short. As the vessels operate within the same area, we can assume that the weather conditions that each individual vessel experiences are rather similar, as they are all in close proximity. Additionally, any change in the weather conditions during the vessels voyage will be minimal, due to short sailing distances. Based on this, we assume in this thesis that all vessels experience the same weather, and that the weather is constant for the whole simulation. This allows us to sample the metocean data only once for each simulation, and the weather sampled will be representative for the entire operational area and valid for the entire fleet. This can, however, not be assumed for operation where fluctuation in the weather conditions should be expected, such as for long sailing distances.

#### 4.1.6.2 Sampling Metocean Data for Long Sailing Distances

For long sailing distances, the weather conditions would be assumed to change throughout the voyage. If that is the case, it is recommended to divide the total sailing distance into shorter legs, and update the weather conditions for each individual leg. It is here important to ensure correspondence between each of the sailing legs, as the weather conditions would be correlated and typically not go from harsh weather to calm waters all of a sudden. Correlation can be ensured by using a state-transition matrix, or Markov chain, where the weather conditions of the next sailing leg are based on the current leg, and changes occur following a defined probability. As an example one could define the sea conditions as calm, moderate and harsh, and use a state-transition matrix as seen in table 10.

Table 10: State-Transition Matrix for Weather Conditions

	<b>Calm</b>	<b>Moderate</b>	<b>Harsh</b>
<b>Calm</b>	0.4	0.5	0.1
<b>Moderate</b>	0.4	0.4	0.2
<b>Harsh</b>	0.1	0.6	0.3

If we assume that the current weather is calm, then there is a 40 % probability that the weather remains calm for the next sailing leg, a 50 % probability that we will face moderate conditions, and a 10 % probability that the weather conditions will be harsh.

As the sailing distance in the case studied in this thesis was relatively short, we could assume that changes in the weather conditions would be minimal, and thus not needed to be accounted for in the model. Consequently, we will not go any deeper into the use of Markov-chains and state-transition matrixes for simulating sea states in this thesis. However, one should be aware of the challenges if attempting to simulate sailing for a longer voyage, and there exists other papers that covers this topic, such as Cristopher J. Green's *Modeling a Weather Environment* (1968).

#### 4.1.6.3 Addition to Sailing Time

The vessel's sailing time is dependent on weather conditions. To account for that, wave-dependent sailing time is introduced in the simulation model. Vessels operating on schedule often sail at designed speed and need to increase power margin up to 15 - 30% to compensate for rough sea conditions in contrast to calm water operations (Gribkovskaia, 2014). However, the vessels sail at maximum speed when responding to an emergency, and thus cannot increase the power output to compensate for harsh weather. Furthermore, when returning from response operations, the vessels sail at economic speed, and seek to minimize the fuel consumption. And so, neither in this situation would it be desirable to increase power margin in order to maintain speed. Consequently, deteriorating weather conditions will lead to an increase in sailing time, while the power output remains constant. Additionally, this increase in sailing time will further lead to a corresponding increase in fuel consumption, as the consumption is given as a function of sailing time.

In this model, we account for the significant wave height,  $H_s$ , as this is the main contributor to sailing time. The contribution by wave height to the sailing time is multiplied as an exponential factor, as seen in formula (i).

$$Sailing\ Time = \frac{Distance}{Speed} \cdot 1.02^{H_s} \quad (i)$$

Going in depth of the implementation of metocean data has not been a priority in this thesis, as this is likely to be altered in a later stage to better correspond with the model developed by MARINTEK. Nonetheless, weather-dependency is an important aspect of marine operations, and was therefore chosen to be included in the model, although without devoting too much time and effort. Additionally, there already exists other papers that study the simulation of weather and its effect on sailing times and other operations, such as Maisiuk and Gribkovskaia (2014).

## 4.2 Mapping and distance calculations

A map projects the surface of a sphere to a flat surface, and thus distortion is inevitable.

There are different types of projections which each have their respective pros and cons, but the projection most frequently used for nautical purposes is the *Mercator projection*, due to its ability to represent lines of constant bearings, known as rhumb lines. As both meridians and parallels are expanded with the same ratio with increased latitudes, all angles are conserved (Bowditch, 2002). Thus, on a map with Mercator projection, drawing a straight line will draw a rhumb line, and so the navigator needs only to set the course once. However, the shortest path between two points along the surface of a sphere is called the great circle. The equator and all meridians are great circles, which implies that the great circle and rhumb line distances are the same on an East-West passage along the equator, and on a North-South passage. Though, as the Mercator projection preserves the angles relative to the meridians, the distortion of areas increases with the latitudes going from the equator towards the poles (e.g. making Greenland appear huge), as seen in figure 5 where all circles have the same area (Støwer, 2015).

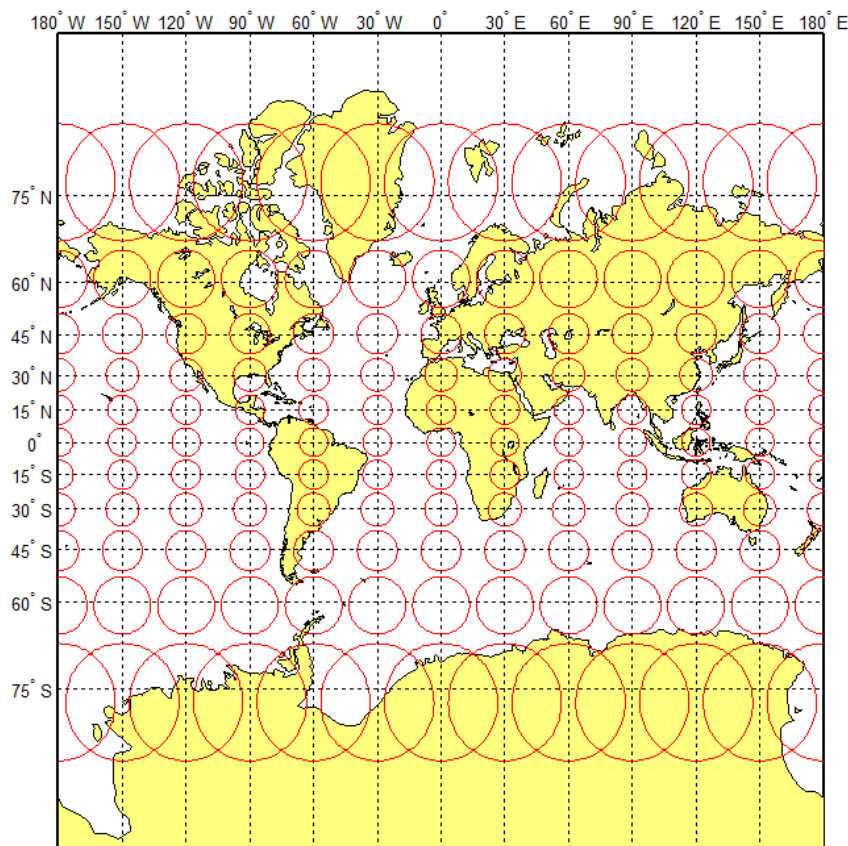


Figure 5: Mercator Projection Showing the Distortion of Areas Further Away from the Equator (Støwer, 2015).



Thus, at higher latitudes, such as the Arctic, the great circle route is significantly shorter than the rhumb line between the same two points, and is the preferred choice when opting for the fastest possible response to an emergency (Bowditch, 2002).

The coordinates of the vessels, the port and the oil spill are all available from the input data, and the following distances are calculated for each vessel using the great circle route, shown in table 11:

*Table 11: Distances of the great circle routes*

<b>Distances of the great circle route from:</b>
Vessel position to oil spill site
Vessel position to port
Port to oil spill site

These distances were calculated using the mapping toolbox in MATLAB, which finds the distance between two points on a sphere using either a great circle arc or a rhumb line arc as defined by the user. This distance is converted from degrees to nautical miles using the MATLAB function *deg2nm*. The reader is referred to *Appendix A: The Simulation Model (MATLAB Script)* for the details of the script.

Knowing these distances, we can calculate the various sailing times between these locations, and also the fuel consumed for each sailing leg.

### 4.3 Calculating Sailing time

The sailing time is calculated as distance divided by speed and expanded with any contributions from the metocean data, as defined in *chapter 4.1.7.3 Addition to Sailing Time*.

$$Sailing\ Time\ to\ Oil\ Spill = \frac{Distance\ to\ Oil\ Spill}{Maximum\ Speed} \cdot 1.02^{H_s} \quad (ii)$$

As mentioned above, the distances for each sailing leg is calculated from the latitudes and longitudes of the points of interest, and in *chapter 4.1 Input Data* we defined the following speed modes for each vessel, shown in table 12.

Table 12: Speed modes for each vessel

Speed mode
Maximum speed
Service speed
Economic speed

The service speed is used for normal day-to-day operation, but when responding to an emergency, we seek to minimize the response time, and thus the vessel will sail at maximum speed. Certain fuel criteria are set in *chapter 4.5 Determining Feasibility with Regard to Current Fuel Level*, which checks the feasibility of a vessel responding to an emergency call based on the current fuel level of the vessel and its expected consumption. In order to save fuel, the vessel will sail at economic speed on the return trip after completing oil spill recovery operations, with the aim to increase availability with regards to the fuel criteria. By that, we mean that a lower estimated total fuel consumption gives the ship a higher chance to be eligible when tested against the required fuel level set by the criteria, providing the model with a higher amount of feasible solutions.

#### 4.4 Estimating the Total Fuel Consumption for the Response

In *chapter 4.1 Input Data*, we defined the vessels' fuel consumption for the following modes, as shown in table 13.

*Table 13: Fuel consumption modes*

<b>Fuel consumption modes</b>
Fuel consumption in maximum speed
Fuel consumption in service speed
Fuel consumption in economic speed
Fuel consumption in operation

As the fuel consumption is given as a function of time, we need only to multiply with sailing time in order to determine the total fuel consumption. This is done for each sailing leg, paired up with the fuel consumption mode representative for that leg. By that we mean that as we decided to sail in max speed when responding to an emergency, the maximum fuel consumption is multiplied with the time it takes to sail to the oil spill site. Likewise, when returning from operation, we use the fuel consumption in economic speed and multiply it by the sailing time in economic speed.

Weather is included through the sailing time, as we already operate in max speed and max fuel consumption during response. Furthermore, as sailing time is increased in harsher weather, the fuel consumption will also be increased as it is given as a function of time.

#### 4.5 Determining Feasibility with Regards to Current Fuel Level

When determining the expected response time of a vessel, one must make sure that the chosen solution is indeed feasible. Earlier in this chapter we mentioned that both the vessel's current operation, the cargo it transports and the emergency equipment installed would impact, and potentially conflict with, the emergency response. In addition to this, it is necessary to evaluate the expected fuel consumption of responding to an emergency, in order to be certain that the vessel has enough fuel onboard to carry out the response task.

Depending on the current fuel level, the model allows for three possible options, listed in table 14:

*Table 14: Response Options with Regards to Fuel*

<b>Response options</b>
The vessel sails directly to the emergency site and has no need for refueling
The vessel sails directly to the emergency site but requires refueling offshore by another vessel
The vessel refuels at port before sailing to the emergency site

##### 4.5.1 Direct Response by an Individual Vessel

In the first scenario the vessel sails directly to the location of the oil spill, conducts the recovery operation and returns to port. The vessel has sufficient fuel available for the operation as a whole, and is consequently in no need for refueling prior to, or during, the operation.

In the model, the vessel is eligible for such a response if the expected fuel consumption for the whole operation is less than or equal to the current fuel level, with a safety factor of 20 %. This includes sailing from the current position to the oil spill in maximum speed, battling the oil spill and returning to port in economic speed. The safety factor is included as an extra precaution to allow for some fluctuation due to any unforeseen events or changes in weather, as well as to have a general buffer planned for the fuel.

#### 4.5.2 Direct Response Requiring Refueling Offshore

The second scenario utilizes two vessels, we may call them vessel A and vessel B, and occurs if the responding vessel (vessel A) does not fulfill the previous fuel requirement; that is, the vessel's current fuel level is below the threshold of the expected total consumption with a safety margin of 20 %. Consequently, the vessel is in need of refueling, and a second vessel (vessel B) is utilized to aid in this manner. While vessel A sails directly to the oil spill and initiates recovery operations, vessel B first travels to port in order to refuel, and then sails to the oil spill to provide fuel for the already operating vessel A. The criterion is formulated as the fuel consumed by vessel A as it sails to the oil spill site, plus the fuel consumed in operation until vessel B arrives for refueling.

$$(Fuel\ Consumption_A^{Max} + Fuel\ Consumption_A^{Operation}) \cdot SF \leq Fuel\ Level_A^{Current} \quad (iii)$$

Where,  $SF$  is the safety factor of 20 %.

The time it takes for vessel B to get to the oil spill site will be the sum of the time spent sailing to port, refueling in port and sailing to the oil spill site. Also, the time vessel A spends on sailing to the oil spill is subtracted, as both vessels responds simultaneously. By that we mean that while vessel A sails to the oil spill site, vessel B will also have started sailing towards the port. Consequently, as we seek to estimate the time vessel A spends in operation mode, the sailing time of vessel A must be accounted for in the total expected response time of vessel B.

$$Response_B = (Sailing\ Time_B^{To\ Port} + Refueling\ Time_B^{Port} + Sailing\ Time_B^{To\ Oil\ Spill}) \quad (iv)$$

$$Time\ in\ Operation_A = Response_B - Sailing\ Time_A^{To\ Oil\ Spill} \quad (v)$$

The time it takes to refuel in port is estimated by how much fuel is needed in order to fill up the fuel tanks, the rate of which the fuel is loaded, and an additional delay caused by navigation in and out of the port.

$$Refueling\ Time = \frac{(Fuel\ Cap - Fuel\ Level^{Current} + Fuel\ Con^{Sail\ to\ Port})}{Refueling\ Rate} + Delay \quad (vi)$$

Where,  $Fuel\ Cap$  stands for fuel capacity and  $Fuel\ Con$  for fuel consumption

In this scenario, a vessel's ability to respond to an emergency call is evaluated together with each and every other vessel of the fleet, as the feasibility of the solution is dependent on the compatibility of two vessels to work together. So, each individual vessel is checked against the fuel criteria, and then all the remaining vessels are tested against that vessel to determine if the two ships can provide a feasible solution through combined efforts.

#### 4.5.3 Refueling in Port

If the vessel does not have sufficient fuel to commit to the oil spill response neither by itself nor by receiving fuel support by a second vessel, it will have to return to port and refuel before sailing out to the oil spill. This would be the same response time as the vessel that acts as a fuel delivery vessel in the previous scenario. Similarly, the response time is estimated as the sailing time to port plus the time it takes to refuel, and the sailing time out to the oil spill, as in the previous scenario.

### 4.6 Determining a Feasible Response

The fastest feasible response plan is determined for each vessel of the fleet by running through the decision tree shown in figure 6.

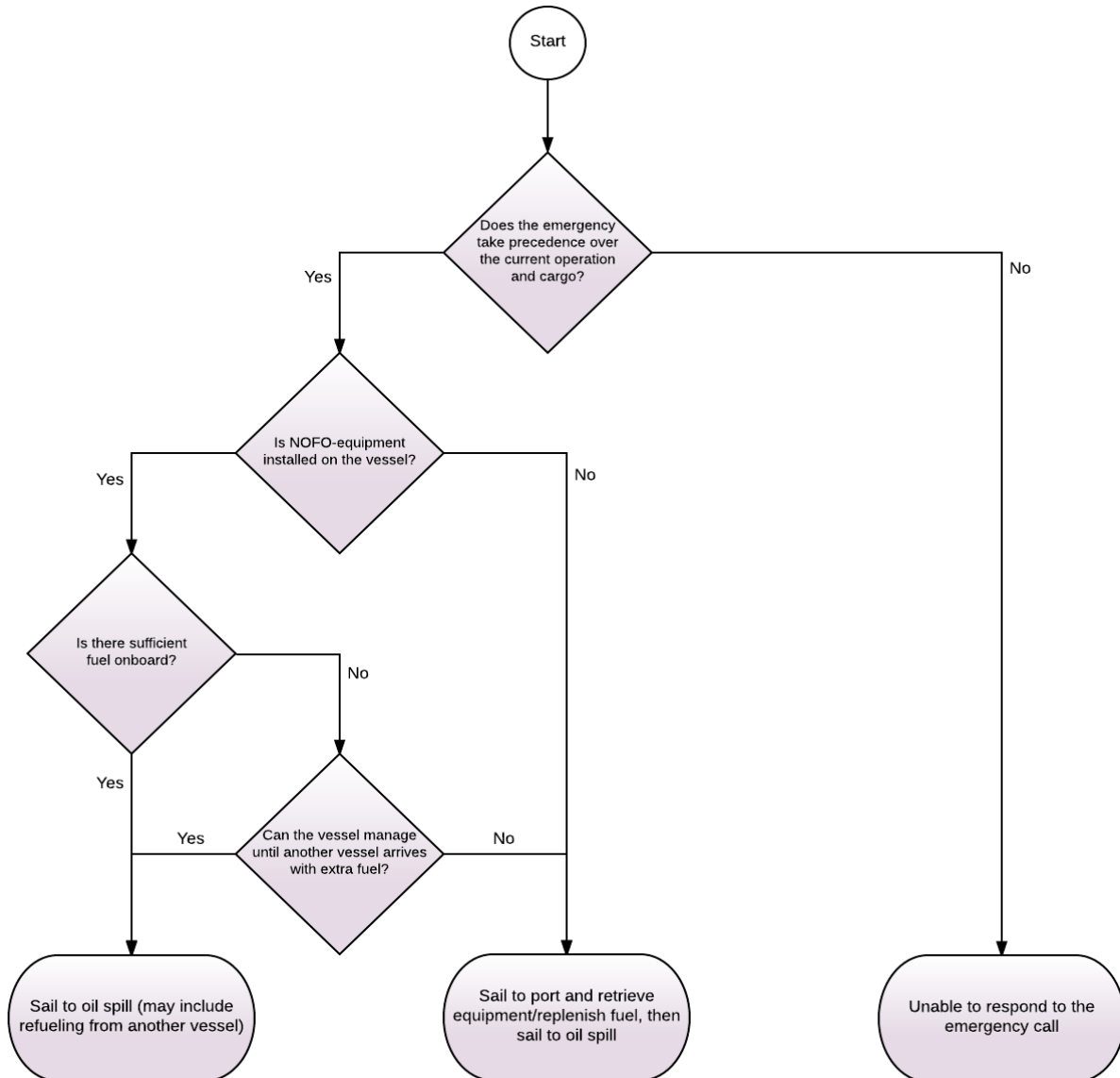


Figure 6: Decision Tree – Deciding a Feasible Response Action for Each Vessel

The model reviews the achieved response times for all vessels, and the solution that provides the shortest response time is chosen.

The solution is then evaluated in terms of how the response will be carried out; Will the vessel sail directly to the emergency site, or does it require supplies from port first? Alternatively, does the chosen vessel require fuel support from another vessel in the fleet, and if so, which vessel is set to this task?

When the nature of the response is determined, the chosen solution is plotted, showing the vessels included in the response and their respective route. This is illustrated in figure 7.



Figure 7: Plotted Solution Example

In this particular example, the solution includes the use of a second vessel as fuel support, which first sails to port and refuels before coming to the aid of the chosen vessel. This vessel's current position is shown by the green circle, and its route by the green lines. The chosen response vessel sails directly to the oil spill area and starts containing the oil spill, shown by the red markers. The third vessel, shown by the black cross, does not have any particular assignment in the initial response.



## Chapter 5

### Case Study: Response Times for a Varying Number of NOFO-Equipped Vessels

During drilling operations, it is paramount to establish some response apparatus that can react to any hazardous contingencies within a reasonable amount of time, in our case, oil spills.

This is commonly solved by having a dedicated standby vessel positioned close to the platform, which is tasked with emergency response and monitoring the safety zone around the rig for potential leaks.

The drawback is that chartering such a vessel is costly, and it could be economically beneficial to eliminate the need for this extra vessel by establishing a sufficient emergency preparedness system through other means. Thus, we want to investigate what oil spill response time can be achieved through alternative solutions.

In order to combat oil spill, one need to have specialized equipment onboard the vessel, following standards set by NOFO. In this case study we look into the option of installing such equipment onboard the supply vessels utilized in the daily operation. However, oil spill response equipment requires a certain deck area when permanently installed on a vessel. Thus, for the logistical aspect of marine operations, it is preferable to have the oil spill equipment stored onshore and retrieve it in case of an oil spill, so that the full deck area can be utilized for transporting cargo in the daily operations. Although, in terms of safety it would be ideal to have every vessel equipped with oil spill response gear readily available onboard. Consequently, we have a trade-off between two aspects of the operation which forms the basis of this case study, where we seek to determine how the fleet response time changes by the number of vessels carrying oil spill response equipment onboard.

We can study this by using the developed model to simulate the oil spill response time of the fleet, and develop a statistically expected response by running a large number of simulations. Furthermore, the simulations can be repeated for several cases, where the number of vessels carrying emergency response equipment can be varied for each case. This allows us to determine how the response time can be expected to change depending on the number of vessels with emergency equipment onboard, which serves as important input when deciding how many vessels are necessary to equip with oil spill response equipment in order to fulfill any safety demands. In short, we seek to equip as few vessels as possible, while still meeting the safety requirements, so that as much deck area as possible of the fleet can be utilized for cargo transport.

We assume a homogenous fleet of three platform supply vessels, with characteristics as defined in *Appendix C: Input Data Sheet*, and run simulations for the four cases as described in table 15, using the model as given in *Appendix D: Case Study Simulation Model*.

*Table 15: The Four Cases Simulated*

<b>Case Number</b>	<b>Description</b>
Case 1	All three vessels have NOFO-equipment permanently installed onboard
Case 2	Two vessels have NOFO-equipment permanently installed onboard
Case 3	Only one vessel has NOFO-equipment permanently installed onboard
Case 4	NOFO-equipment is not installed on any of the vessels, and must be retrieved from port in case of an oil spill

For each case we perform ten thousand simulations and register the minimum response time for each simulated scenario. We then plot the number of occurrences of the registered response times, and also the probability that the response time is below a certain threshold. Lastly, the mean response time is calculated for each case.

### 5.1 Stochastic Input Data

In this case study we utilize the model as described in *chapter 4: Model Applied for Simulation of Oil Spill Response Time*, with some minor changes in order to run the model for many repeated simulations. Besides adding a loop to perform the desired number of simulations, the only changes done is that some input data is defined as stochastic rather than deterministic, and thus vary randomly within a given interval. This is due to the fact that we try to simulate many scenarios, where certain parameters will vary between each simulation run, and we do not have deterministic data for all ten thousand scenarios available. However, we know within what range these variables can be assumed to be, and thus allow the variables to be chosen within said interval. As an example, it would be strange to assume that all vessels are in the exact same location every time an oil spill occurs, as they travel back and forth between the port and platform. Yet, it is reasonable to assume that the vessels are located somewhere along this route.

The stochastic variables are given in table 16, and how they are defined in the model is presented in the following sub-chapters.

*Table 16: Stochastic Input Data*

<b>Stochastic Input</b>
Current Fuel Level
Operational Status
Cargo Status
Vessel Location
Metocean Data (Date)

### 5.1.1 Current Fuel Level

The day-to-day fuel level of a vessel is assumed to be uniformly distributed between 50 and 300 [tonne], based on information provided by MARINTEK.

### 5.1.2 Operational Status

In *chapter 4.1.4.2 Operational Status* we defined the operational status variable as an integer, where the numerical value represented the current operational status of the vessel, as described in table 17 below. The operational status of each vessel is chosen randomly between the different options provided, based on the assumed probability of occurrence of each option in table 17. It is assumed a 2 % probability of being unable to abort the current operation, and consequently be unable to respond to the oil spill emergency. Furthermore, it is assumed an equal probability of whether the vessel is sailing towards - or returning from the platform. However, the case of *port navigation* cannot be chosen randomly as the other instances, as this is dependent on vessel location. As an example, it would not make sense to set the operational status as *port navigation* if the vessel is sailing in open sea in no close proximity to the port. This is accounted for by setting the operational status variable as *port navigation* if the latitude of the vessel is within 0.05 decimal degrees of the port location.

*Table 17: Choosing the Operational Status Variable*

<b>Variable Value</b>	<b>Definition</b>	<b>Probability/Criteria of Being Chosen</b>
0	Unable to respond	2 %
1	Port navigation	Within a 0.05 decimal degree proximity of the port
2	Sailing towards offshore installation	49 %
3	Returning from offshore installation	49 %

### 5.1.3 Cargo Status

The cargo status was defined as a binary variable declaring whether the current cargo interfered with the vessels ability to respond to an oil spill emergency or not. It is assumed that having cargo that takes priority over, or interferes with, the emergency response, is a very rare event, and is thus modeled with a probability of one in a thousand. This is implemented by choosing a random number in the interval 1 to 1000. If the number is 1, then the cargo status variable is set to 0, meaning the vessel is unable to respond to the oil spill emergency. However, if the random number drawn is any other number in the given interval, the variable is set to 1, and the vessel is eligible for emergency response.

### 5.1.4 Location

The route from port to oil spill is divided into three sailing legs, in order to be able to assign random vessel positions that more accurately follow a realistic route. This is done by dividing the latitudes and longitudes of the voyage into three areas, defined by the three *thirds* constituting the total coordinate range of the voyage, illustrated in figure 8. In other words, the area closest to the port will have a defined range where the lower limit is the coordinates of the port, and the upper limit is one third of the distance to the oil spill in both latitudinal and longitudinal direction, and so on. We would like to remind the reader that the platform location is used for the assumed location of the oil spill, which is why the vessel route is given between the oil spill and the port.

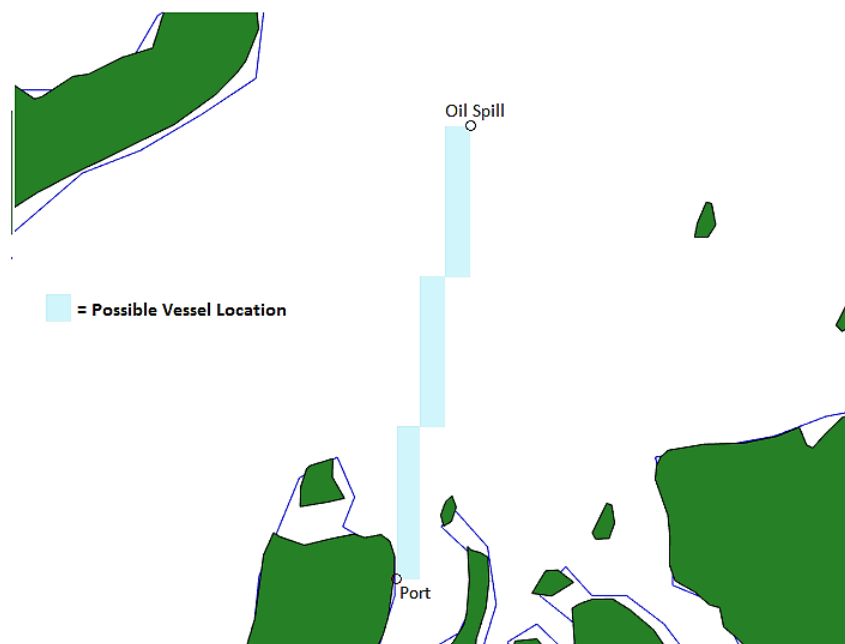


Figure 8: The Sailing Route Divided into Three Rectangular Areas Where the Vessels Are Assumed to Operate

By dividing the sailing route into these three areas, we limit the options of which the vessel positions can be defined to a more representative route.

If we had simply allowed the coordinates of the vessel to be randomly generated within the range of the coordinates of the port and oil spill, the model would be exposed to scenarios where the location of the vessels could be in places far off the real route, or even on land, as shown by the red markers in figure 9.

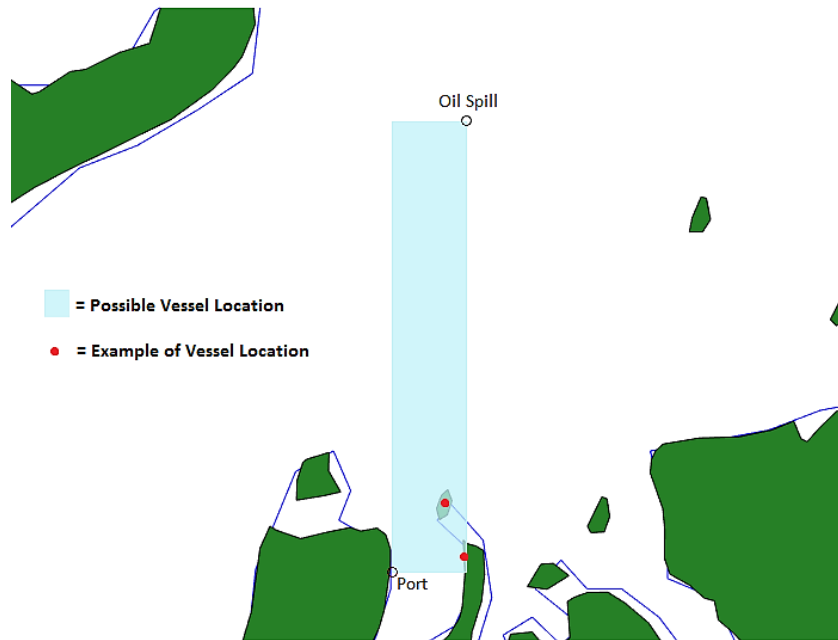


Figure 9: The Sailing Route Defined by Only One Rectangular Area

In the model, the latitude of the vessel is chosen randomly between the port and the oil spill, and further checked to see within which of the three defined areas of figure 8 it is located. This area, determined by the given latitude, will further define the upper and lower limit of the longitude, which is then chosen randomly within the defined range.

It has not been prioritized to put an extensive amount of work into representing more realistic sailing patterns, as this will be provided by MARINTEK's model for future simulations.

#### 5.1.5 Metocean Data

We assume that there are no seasonal limitations, so that the vessels are in operation the entire year. The date for which the emergency can occur is thus picked at random from a uniformly distributed set of months ranging from 1 – 12 and days ranging from 1 – 28/30/31. We have not included leap years, as this doesn't affect the weather conditions in any other way than normal day to day variations, and provides limited contribution to the data set as a whole. The reason that we still choose to differentiate between months is due to the fact that not doing so would limit each month to 28 days, as a higher number of days per month would potentially lead to an invalid data set for February. This would consequently result in the exclusion of a lot of weather data, and the validity of the model output would suffer.

## 5.2 Results of the Case Study for a Varying Number of NOFO-Equipped Vessels

As mentioned in the introduction of the case study, we run ten thousand simulations for each of the following four cases:

- Case 1: All three vessels are installed with NOFO-equipment
- Case 2: Two vessels are installed with NOFO-equipment
- Case 3: Only one vessel is installed with NOFO-equipment
- Case 4: No vessels are installed with NOFO-equipment

For each case we register the minimum feasible response time for all ten thousand simulations. We then use these values to create a probability density plot and a probability plot. As to easier distinguish between the two plots in the text, we will refer to the probability density plot as the density plot.

The density plot indicates the relative probability of the various response times, given in 0.5 hour intervals. This is determined by the frequency a given response time occurs throughout all the simulations.

The probability plot shows the cumulative probability with increasing response time. This allows us to determine the probability that the response time will be below a given amount of hours, and thus estimate the upper bound with high certainty.

As a matter of good scientific practice, a significance level is chosen for the data collection. If within the significance level, one may include that the observed results are representative for the study, and is not caused by an error or unrealistic conditions. The probability plots are presented with a 1 % significance level in the figures, but also a 5 % significance level will be commented. In the density plot, all measured response times are included.

Lastly, the mean response time is calculated for each of the cases and summarized in table 19 at the end of this chapter.



### 5.2.1 Three Vessels Installed with NOFO-Equipment

For the first case, all vessels of the fleet have oil spill response equipment permanently installed. This will provide the fastest response, and is ideal in terms of safety. In figure 10 the density plot is presented, illustrating the relative rate of occurrence of the different response times.

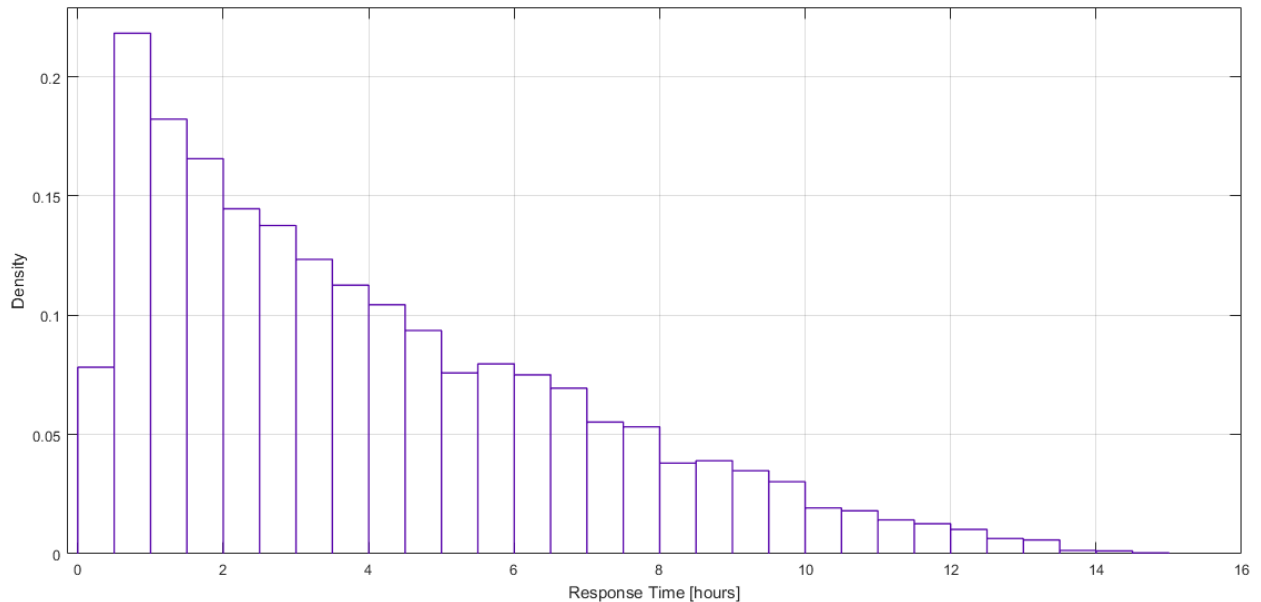


Figure 10: Density Plot of the Response Times - Three Vessels Equipped

During the ten thousand simulations runs, response times were registered in the span from <0.5 to about 14.5 hours. Sixty percent of the response times are measured within the first four hours, and gradually decline as the number of hours increases. There are no extreme isolated measurements taking place way above the main bulk of registered times, as all three vessels are properly equipped for combating oil spill, providing each simulated scenario with at least one decent feasible solution. This is further witnessed in the probability plot in figure 11.

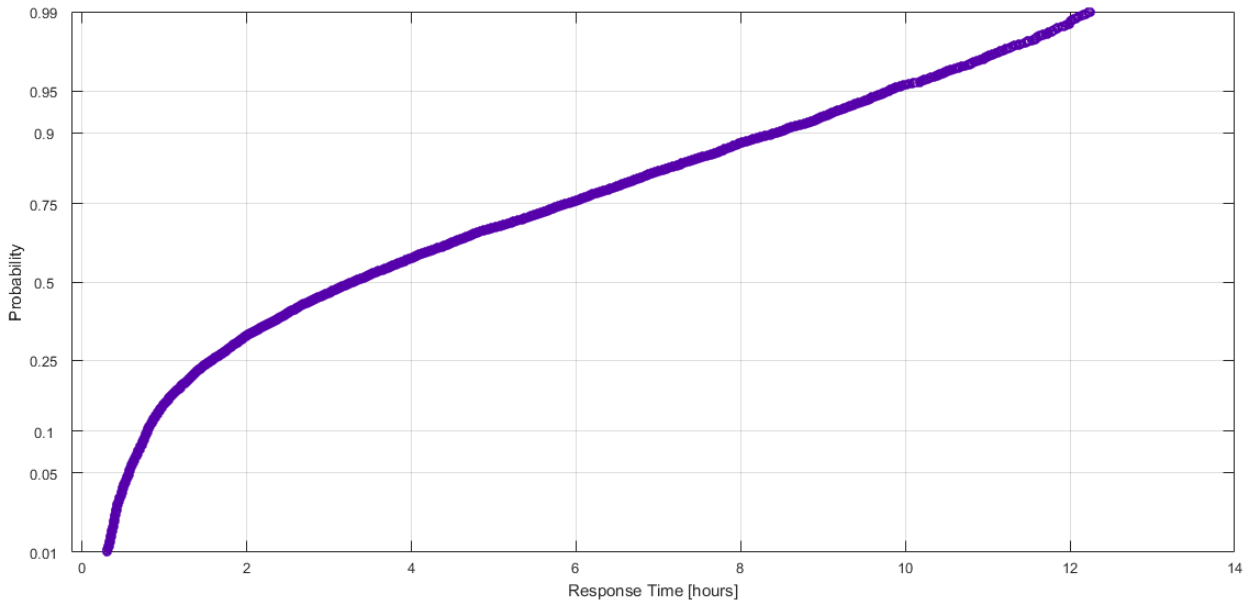


Figure 11: Probability Plot of the Response Times - Three Vessels Equipped (0.01 Significance Level)

Figure 11 shows that the response time has a 0.99 probability of being below 13 hours and a 0.95 probability of being less than 10 hours. A 0.99 probability of being below 13 hours is close to the maximum response time measured across all simulations, telling us that the worst case scenario of about 14.5 hours does not deviate much from the maximum response within the 1 % significance level. This cannot, however, be said to hold true for the second fleet composition evaluated.

### 5.2.2 Two Vessels Installed with NOFO-Equipment

In the second case, the NOFO-equipment is removed from one of the vessels while maintained on the remaining two. This allows for more cargo to be transported on each trip, but will weaken the safety function of the fleet. This is shown in figure 12, where the density plot reveals some extreme incidents with registered response times up to 34 hours.

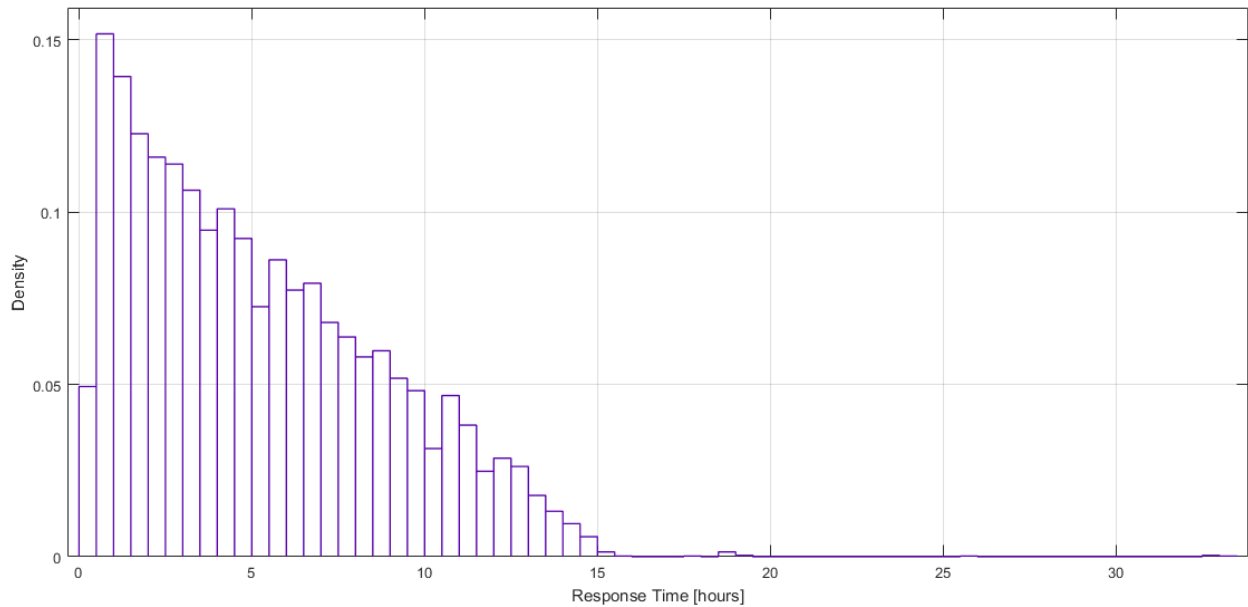


Figure 12: Density Plot of the Response Times - Two Vessels Equipped

The majority of the results are still centered around the first few hours, and gradually declining until roughly 15 hours. However, past this point we now witness some scenarios deviating from the norm, with a measured response in the range of 16 to 34 hours. On the other hand, the probability plot in figure 13 reveals that these results are not included within a 1 % significance level, and are thus rejected as statistically representative of the expected response time of the fleet.

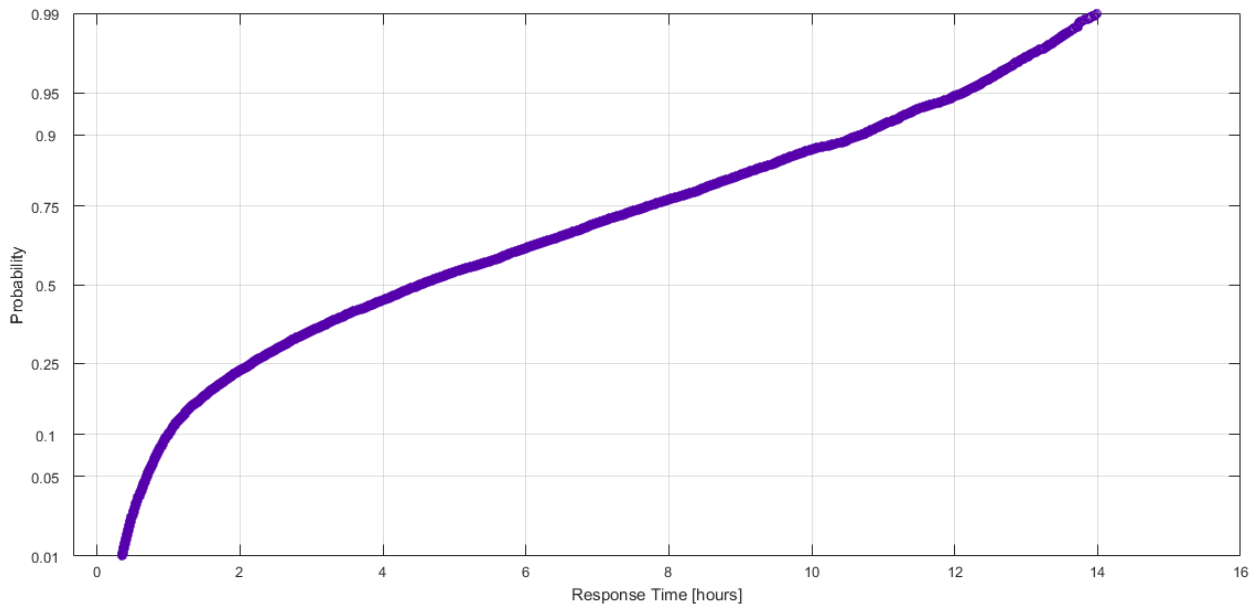


Figure 13: Probability Plot of the Response Times - Two Vessels Equipped (0.01 Significance Level)

This is a perfect example of why we choose to look at the probability plot for a certain significance level. In this case we have measured some single incidents where the response time was as high as 34 hours. However, the probability plot in figure 13 shows that the response times in the 95<sup>th</sup> and 99<sup>th</sup> percentile will be no more than 12 and 14 hours, respectively. Hence, any incidents where the response takes longer than e.g. 14 hours can be thought of as isolated incidents that do not properly reflect the viability of the preparedness system. This is due to how some input data is drawn randomly from within a specified interval, which allows the worst case scenario of every parameter to potentially coincide. E.g. all vessels are positioned by the platform, and all are low on fuel. Such incidents are highly unlikely to occur in real life operations, and gives a poor representation of system performance. The feasibility of the emergency preparedness should therefore be validated based on a 1 % or 5 % significance level, as this better represents what one likely can expect as an upper bound when equipping two vessels with NOFO-equipment.

### 5.2.3 One Vessel Installed with NOFO-Equipment

In the third case we witness that the distribution of response times deviates from the previous reclining-slope, as seen in figure 14.

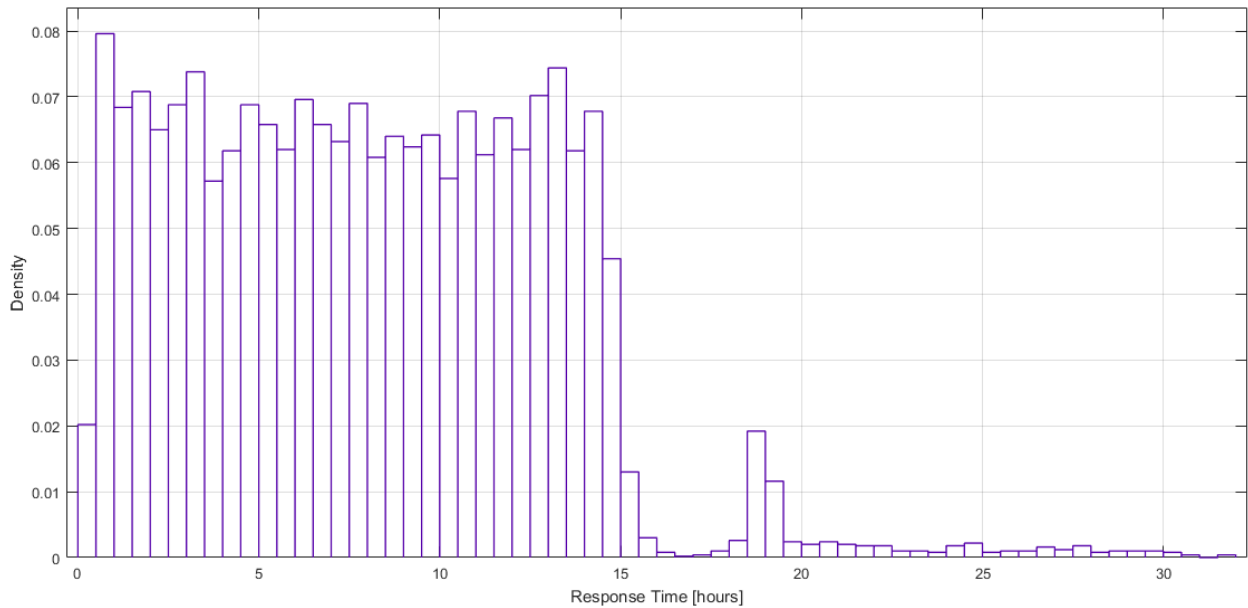


Figure 14: Density Plot of the Response Times - One Vessel Equipped

Most results are still in the interval from 0.5 to 15 hours, but more evenly distributed within this range. Additionally, the occurrence of measurements from 16 to 34 hours is more frequent than in the previous case. Furthermore, in figure 15, the 99<sup>th</sup> percentile now includes response times up to about 23 hours. However, there is still a 0.95 probability for the response time to be less than 15 hours.

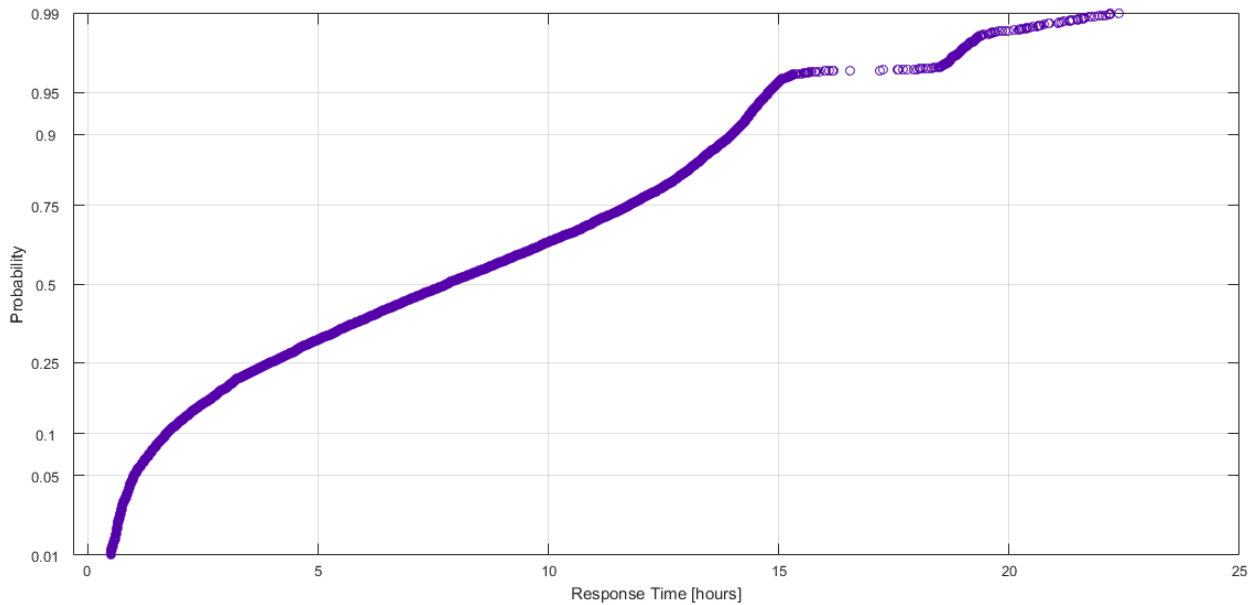


Figure 15: Probability Plot of the Response Times - One Vessel Equipped (0.01 Significance Level)

There can be witnessed a small gap in the graph from 16 to a little over 17 hours. This is where the feasible solution shifts from direct response to requiring a stop in port. This gap occurs naturally as the upper bound of direct response times does not overlap with the lower bound of the response by port. A high direct response time would suggest that the vessel is located close to port and must sail the full length of the route in order to arrive at the oil spill site. Sailing this distance can take up to 16 hours depending on the weather, while from figure 17 in the next case we can see that a response by port is expected to require about 17 hours as a minimum. This is logical because for both the maximum direct response and the minimum response by port, the vessels will sail roughly the same distance. However, the latter also requires additional time in port in order to load the NOFO-equipment onto the vessel.

### 5.2.4 No Vessels Installed with NOFO-Equipment

For the final case, no vessels have oil response equipment installed onboard. Consequently, the vessels always have to stop in port and retrieve the necessary equipment before sailing to the oil spill and initiate oil recovery operations. This causes a jump in the lower end of the registered response times, which in figure 16 are now shown to be in the range of 16.5 to 33 hours.

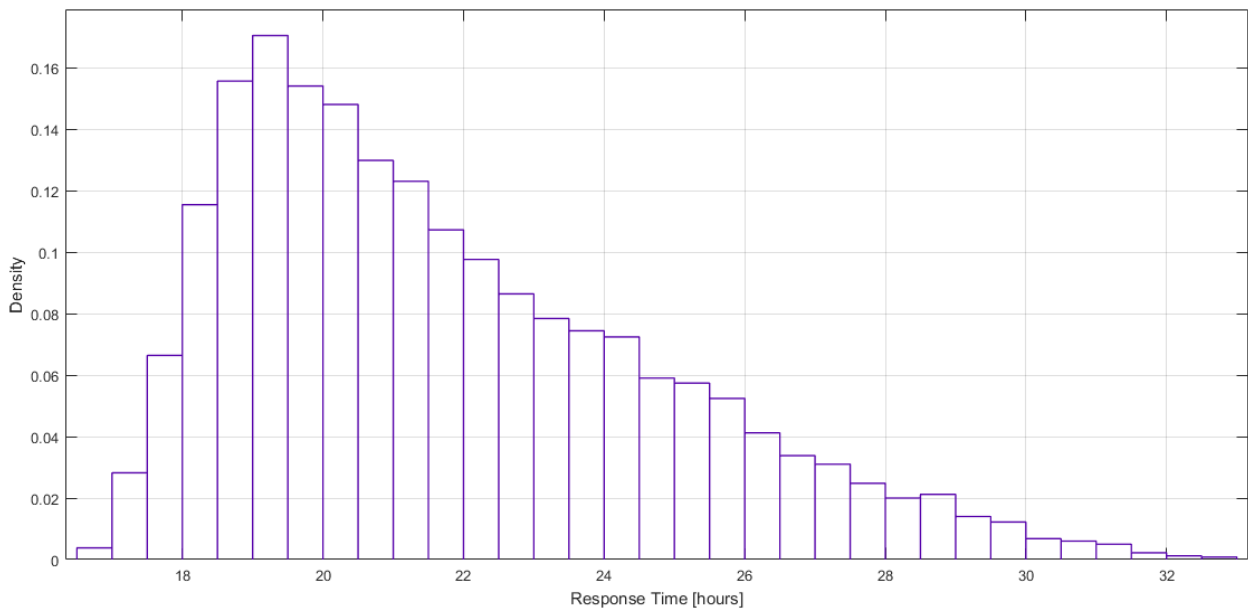


Figure 16: Density Plot of the Response Times - No Vessels Equipped

We see that the density plot has returned to the usual shape, yet stabilized at a much higher number than previously. As all vessels must retrieve NOFO-equipment from port, we no longer have the fluctuation in response times caused by often being able to provide direct response, yet regularly having to first stop by the port. As for the previous cases, the maximum response time is determined from the corresponding probability plot, here given in figure 17.

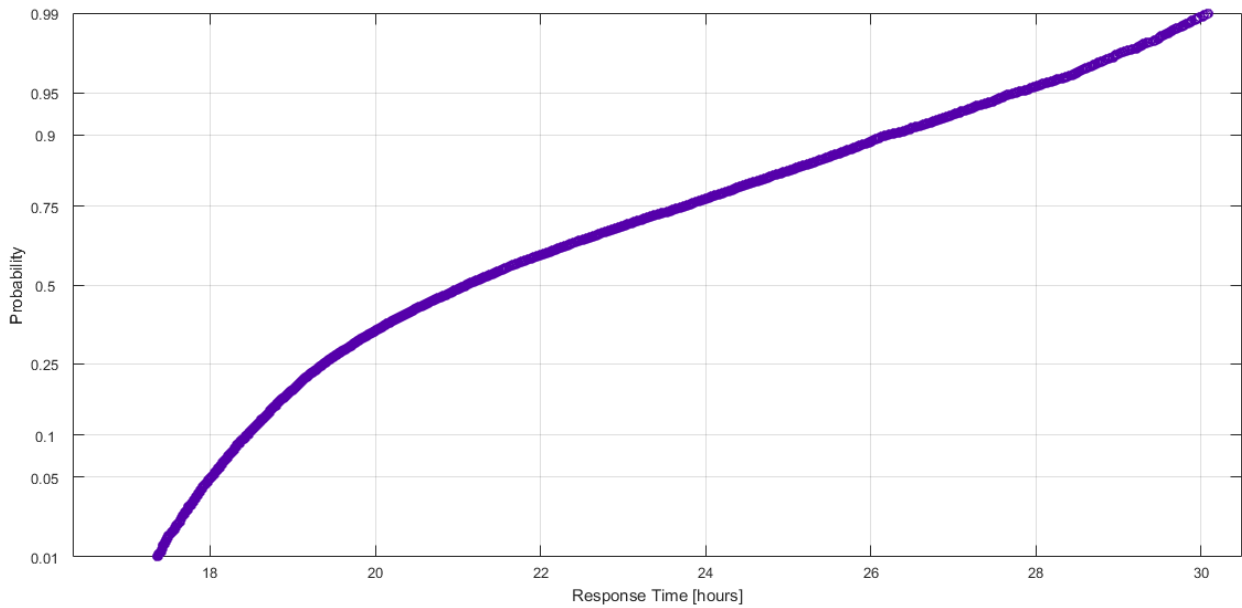


Figure 17: Probability Plot of the Response Times - No Vessels Equipped (0.01 Significance Level)

For 99 % of the simulations run, the response time is found to not exceed 30 hours, and for 95 % the simulated response time would be no more than 28 hours.

In order to prove that we can achieve a satisfactory response function through our solution, we need to determine both the expected performance and an upper boundary, which with a high certainty can be guaranteed to not be surpassed. These results are summarized in the two tables below for all four investigated cases, where table 18 presents the maximum response time for a 1 % and 5 % significance level, respectively, and table 19 presents the mean response time found for each evaluated case.

Table 18: Maximum Response Time Within a 1 % and 5 % Significance Level

<b>Number of Vessels with NOFO-Equipment Available Onboard</b>	<b>Maximum Response Time (1 % Significance Level) [hours]</b>	<b>Maximum Response Time (5 % Significance Level) [hours]</b>
3	12.2	9.7
2	14.0	12.0
1	22.5	14.5
0	30.0	27.7



The maximum response time does not change drastically in the first three cases for the 5 % significance level, but rather increases with about two and a half hours for each NOFO-equipment uninstalled. However, the maximum response time takes a big jump already between the second and third case when evaluating based on a 1 % significance level. This is due to a more frequent occurrence of response times in the higher end of the spectrum for the third case, which is not included in the 0.95 interval. The impact of these high measurements are also reflected to some degree in the mean response time in table 19, which increases by nearly three hours by the removal of the second set of NOFO-equipment.

*Table 19: Mean Response Time for the Four Cases Simulated*

<b>Number of Vessels with NOFO-Equipment Available Onboard</b>	<b>Mean Response Time [hours]</b>
3	4.0
2	5.2
1	8.1
0	21.8

The mean response time only increases by little over an hour from the first to the second case, as both these two cases almost exclusively utilized direct response. As mentioned, from the second to the third case the mean response time increases by nearly three hours, as there was a higher occurrence of responses traveling by port, but more importantly, because only one vessel was equipped with proper gear. Sailing to the oil spill directly will always provide the fastest response, so this single vessel was always chosen if possible, but compared to the previous two cases who can rely on several vessels, this vessel would naturally be more often located at unfavorable positions. For the last case, we witness a big increase in the mean response time. This was expected, as all solutions now require the vessel to first make a stop in port to retrieve the NOFO-equipment.

### 5.3 Conclusion of the Case Study

All in all, this case study was meant to illustrate how the developed simulation model can be utilized to evaluate different fleet compositions with regards to response time and help aid in the decision making when determining the fleet specifications with regards to both emergency preparedness and logistics planning. It is shown that the model can help simulate the expected response time of a fleet, which can further be used to determine whether the fleet provides a satisfactory emergency preparedness function against acute oil spills.

## Chapter 6

### Discussion

The case study proved that the simulation model is able to provide the desired results, and so the focus of this chapter will be on discussing the *validity* and *versatility* of the developed model.

The validity of the simulation output is affected by both the uncertainty of the input data, as well as simplifications and assumptions made in order to represent the real world operations. We cannot 100 % replicate the real world processes. However, we seek to develop a model that is representable for real world behavior, and can provide results that resembles the reality. By producing a model that is representable of the real world, we can analyze and make valid predictions regarding the feasibility of a theoretical solution. Accordingly, the simplifications and assumptions made are necessary in order to be able to describe the real world operations through a simulation model.

With that being said, the greatest source of uncertainty in this thesis is related to the representation of input data in the case study. It was witnessed in the case study that when allowing input data to be randomly drawn within a specified interval, one run the risk of having many unfavorable conditions coincide, resulting in extreme measurements that are very unlikely to occur in real life operations. This was dealt with by neglecting any results that are deemed not statistically significant. However, errors related to the representation of input data will not be a problem in the real project, as input will be provided from MARINTEK. When the daily operation model from MARINTEK is up and running, we will for example be able to produce more realistic sailing patterns, so that the vessel location is better represented based on actual sailing schedule. Furthermore, critical conditions, such as conflicting cargo, lack of fuel etc., will no longer occur for the majority of the vessels within the same simulated scenario.

Regarding the processing of data in the model, and the simplifications and assumptions made in the modeling process, there are a few aspects that can be commented on. First off, if a vessel is not eligible for combatting the oil spill, because it cannot abort its current operation, it was said in *chapter 4.1.4 Fleet Status* to be simply deemed as a non-feasible solution, and taken out of the evaluation. It was not checked for how long it is tied up with its current operation, and the vessel could in theory finish the operation and still be ready to tackle the

oil spill before any other vessel arrives. However, it was assumed that the oil spill would almost always be prioritized over other operations.

Another weakness in the current model is that there are set no operational limitations based on environmental conditions. In reality, harsh weather conditions may affect the vessels ability to perform oil spill response operations, refueling at sea, or even sail. If, for example, the wave height exceeds a certain threshold, some operations may no longer be feasible, and the vessel might even have to wait for appropriate weather conditions before being able to respond to the emergency.

Lastly, we would like to comment on the second fuel criteria, where the possibility of utilizing a second vessel as fuel support is evaluated. In this scenario, the model does not allow for the fuel supporting vessel to sail directly to the oil spill site to refuel the responding vessel, but is forced to first refuel in port. This may cause the loss of some feasible solutions where the responding vessel is dependent on receiving fuel support within a short period of time. Although, in these scenarios, the fuel support vessel is likely to have enough fuel to take over the role as the responding vessel, given that the necessary emergency equipment is installed.

In terms of versatility, the model is applicable for several types of emergency response operations, such as fire accidents and other rescue operations. The simulation model may require some minor adjustments based on the specifics regarding the scenario at hand. However, the same principles and argumentation are still valid: do we have the necessary equipment onboard, is the current fuel level sufficient for conducting the operation at hand, and so on. Subsequently, the minimum response time can be calculated based on sailing time and feasibility of the vessels.

## Chapter 7

### Conclusion and Further Work

#### 7.1 Conclusion

Arctic operation is a trending topic, and the current amount of experience related to marine operations in this area is somewhat limited. Simulation may be a valuable tool when planning offshore operations in the Arctic, as one can develop a simplified model to perform test-runs for the real-life project, and support the decision-making of important parameters in the project, as well as potentially cut the lifetime costs of the project. Thus, the development of simulation models applicable for Arctic operations are valuable for the progression of both simulation as a decision support tool, and the geographical expansion of the maritime industry.

Regarding emergency preparedness, the developed simulation model may help prove that a satisfactory response is achievable through other means than assigning an extra vessel to this task. In particular, that through thorough fleet management and operational planning, one can provide a safe and satisfactory oil spill preparedness function by utilizing the existing vessels in the fleet for response. This could further reduce the lifetime costs of the project, as there would no longer be need for a supplementary standby vessel tasked with emergency response.

In the beginning of the thesis, the objective was formulated as follows.

*The objective of this thesis is to develop a simulation model that can evaluate the expected emergency response time of a given fleet composition, and thus serve as a decision support tool for strategical fleet sizing and operational planning.*

Through the case study, it was shown that the model is indeed capable of simulating the expected emergency response time of a fleet, and is thus able to aid in the decision making when determining the fleet specifications with regards to both emergency preparedness and logistics planning.

## 7.2 Further Work

Further development of this model will be in terms of establishing sufficient communication and sharing of data across platforms with the model developed by MARINTEK. For an external reader that wish to utilize this model, it is recommended to include some operational limitations given by weather conditions, such as wave height, and also extend upon the relation between weather conditions and their effect on sailing times.

---

## Reference List

- Aneichyk, T. (2009). *Simulation Model for Strategical Fleet Sizing and Operational Planning in Offshore Supply Vessels Operations*. Molde University College. Retrieved from [http://brage.bibsys.no/xmlui/bitstream/handle/11250/153445/master\\_aneichyk.pdf?sequence=1](http://brage.bibsys.no/xmlui/bitstream/handle/11250/153445/master_aneichyk.pdf?sequence=1)
- Antonio. (2008). Function for faster data transfer Matlab <--> Excel. Retrieved from MathWorks Community: <http://www.mathworks.com/matlabcentral/fileexchange/22365-function-for-faster-data-transfer-matlab-%3C-%3E-excel>
- Bowditch, N. (2002). *The American Practical Navigator, An Epitome of Navigation*. Bethesda, MD: National Imagery and Mapping Agency.
- Carson II, J. S. (2005). *Introduction to modeling and simulation*. Paper presented at The 2005 Winter Simulation Conference, Orlando, FL.
- Green, J. C. (1968). *Modeling A Weather Environment*. Paper submitted to The 1968 Winter Simulation Conference, New York, NY. Retrieved from [http://www.informs-sim.org/wsc68papers/1968\\_0077.pdf](http://www.informs-sim.org/wsc68papers/1968_0077.pdf)
- Klaussen, A. (2013). *Oil Spill Response – The Norwegian Model*. Presentation for INTSOK's Norway Days in Baku Conference, Azerbaijan. Retrieved from <http://www.intsok.com/content/download/20514/128695/version/1/file/NorLense+OSR+the+Norwegian+Model1Aktuell.pdf>
- Maisiuk, Y., & Gribkovskaia, I. (2014). Fleet Sizing for Offshore Supply Vessels with Stochastic Sailing and Service Times. *Procedia Computer Science*, 31(0), 939-948. doi: <http://dx.doi.org/10.1016/j.procs.2014.05.346>
- NOFO. (2013). *Effektiv og Robust Oljevernberedskap - Dimensjonert etter Operatørens Beredskapsplaner*. Sandnes, Norway: Vikmedia AS.
- NOFO. (2016). Principle Illustration of the Multi-Barrier System. Reprinted from *NOFO*, retrieved May 25, 2016 from <http://www.nof.no/Plangrunnlag/Aksjonsmal-og-miljomal/Strategier-og-metoder1/>

- Norconsult AS. (2010). *Beredskap mot akutt oljeforurensning i nordområdene – Status og forventet utvikling*. Retrieved from <https://www.norskoljeoggass.no/PageFiles/6532/Beredskap%20mot%20akutt%20oljeforurensning%20i%20nordomr%C3%A5dene.pdf>
- Nordbø, H. (2013). *Optimal configuration of supply logistics for remote oil and gas fields*. Norwegian University of Science and Technology. Retrieved from <http://brage.bibsys.no/xmlui/handle/11250/238594>
- Norwegian Oil and Gas Association. (2013). *Veiledning for miljørettede beredskapsanalyser* (4th ed.). Retrieved from <https://www.norskoljeoggass.no/Global/2013%20Dokumenter/Publikasjoner/Veiledning%20beredskapsanalyser%202013.pdf>
- Oceans System Simulation. (2015). Weather data set from Waveclimate.com is access restricted, but was given in the course TMR4565 - Oceans System Simulation during the fall semester at the Department of Marine Technology, NTNU.
- Petroleum Safety Authority. (2005). *Consent for permanent area emergency preparedness Oseberg-Troll*. Retrieved from <http://www.psa.no/news/consent-for-permanent-area-emergency-preparedness-oseberg-troll-article2347-878.html>
- SINTEF Materials and Chemistry. (2014). *The OSCAR Model*. Retrieved from [https://www.sintef.no/globalassets/upload/materialer\\_kjemi/faktaark/environment/oscarfact.pdf](https://www.sintef.no/globalassets/upload/materialer_kjemi/faktaark/environment/oscarfact.pdf)
- Støwer, S. K. (2015). *Simulating Ocean Engineering Systems*. The Norwegian University of Science and Technology.
- Ulstein, M. (2014). *A simulation-based decision support tool for arctic field logistics*. The Norwegian University of Science and Technology. Retrieved from <http://brage.bibsys.no/xmlui/handle/11250/238817>
- Westerberg, V. (2012). *Arctic Oil Spill Response Recovery operations - Management and Performance*. KTH Royal Institute of Technology. Retrieved from <http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A550494>



## Appendix



## Appendix A: The Simulation Model (MATLAB Script)

### A.1 Initialization

```
%Made by David Josefsen, spring 2016, for his master's thesis at the
%Norwegian University of Science and Technology, Department of Marine
%Technology

%% Initialization

%Clear out all data and figures for a clean run
clc
clear all
close all

%Read csv datafile of metocean data into a matrix
Weather = csvread('WeatherData.csv',0,0);

%Initializes the function xlsread1.m which keeps the excel sheet open
%throughout the simulation, instead of opening and closing excel each
%time a parameter is read. This caused tremendous saving in the program
%run time.

%Requires xlsread1.m in the same folder as this program

Excel = actxserver ('Excel.Application');
%NB!
%REMEMBER TO DEFINE THE PATH OF THE INPUT DATA FILE BELOW
File='C:\Users\David\Documents\NTNU\MASTER\MATLAB\INPUTDATA.xlsx';
if ~exist(File,'file')
    ExcelWorkbook = Excel.Workbooks.Add;
    ExcelWorkbook.SaveAs(File,1);
    ExcelWorkbook.Close(false);
end
Excel.Workbooks.Open(File);
```



## A.2 Input Data

```

%% Input Data
%Choose the first row in the data set
InitializeRow = 3;
FirstRow = num2str(InitializeRow);

%Fleet Characteristics

%Vessel data that will serve as input
Counter = 1;
FleetRow = FirstRow;
while isempty(xlsread1('INPUTDATA.xlsx',1,['D',FleetRow])) == 0
Fleet(Counter) = xlsread1('INPUTDATA.xlsx',1,['D',FleetRow]);
Counter = Counter + 1;
FleetRow = str2num(FleetRow);
FleetRow = FleetRow + 1;
FleetRow = num2str(FleetRow);
end
EndRow = num2str(str2num(FleetRow) - 1);
%Number of Vessels
nVessels = length(Fleet);
%Max speed [knots]
SpeedMax = xlsread1('INPUTDATA.xlsx',1,['E',FirstRow,':','E',EndRow]);
%Service speed [knots]
SpeedService = xlsread1('INPUTDATA.xlsx',1,['F',FirstRow,':','F',EndRow]);
%Economical speed [knots]
SpeedEco = xlsread1('INPUTDATA.xlsx',1,['G',FirstRow,':','G',EndRow]);
%Fuel consumption in max speed (14.6 knop) [tonne/hour]
FuelConMax = xlsread1('INPUTDATA.xlsx',1,['H',FirstRow,':','H',EndRow])/24;
%Fuel consumption in service speed (12 knop) [tonne/hour]
FuelConService =
xlsread1('INPUTDATA.xlsx',1,['I',FirstRow,':','I',EndRow])/24;
%Fuel consumption in economical speed (10 knots) [tonne/hour]
FuelConEco = xlsread1('INPUTDATA.xlsx',1,['J',FirstRow,':','J',EndRow])/24;
%Fuel consumption in operation [tonne/hour]
FuelConOperation =
xlsread1('INPUTDATA.xlsx',1,['K',FirstRow,':','K',EndRow])/24;
%Fuel capacity of the fleet [tonne]
FuelCapacity = xlsread1('INPUTDATA.xlsx',1,['L',FirstRow,':','L',EndRow]);

%Fleet Status

%Current fuel level in [tonne]
FuelLev = xlsread1('INPUTDATA.xlsx',1,['M',FirstRow,':','M',EndRow]);

%Operational status
%0 = unable to abort current operation, 1 = in port, 2 = in
%transit going towards platform (or at platform), 3 = return trip
OpStatus = xlsread1('INPUTDATA.xlsx',1,['N',FirstRow,':','N',EndRow]);

%Cargo status
%Does delivery of cargo take priority or conflict with response?
%0 = unable to do oil spill, 1 able to.
CargoStatus = xlsread1('INPUTDATA.xlsx',1,['O',FirstRow,':','O',EndRow]);

%Emergency Equipment Status (0 if unequipped, 1 if equipment is available)
EmergencyEquipment =
xlsread1('INPUTDATA.xlsx',1,['P',FirstRow,':','P',EndRow]);

```

```

%Locations in latitudes and longitudes
%Location of vessels
Latitude = xlsread1('INPUTDATA.xlsx',1,['Q',FirstRow,':','Q',EndRow]);
Longitude = xlsread1('INPUTDATA.xlsx',1,['R',FirstRow,':','R',EndRow]);
%Location of port
LocationPort = [xlsread1('INPUTDATA.xlsx',1,['S',FirstRow]),
xlsread1('INPUTDATA.xlsx',1,['T',FirstRow])];
%Location of oil spill
LocationOilSpill = [xlsread1('INPUTDATA.xlsx',1,['U',FirstRow]),
xlsread1('INPUTDATA.xlsx',1,['V',FirstRow])];

%Refueling rate of the port [tonne/hour]
RefuelingRatePort = xlsread1('INPUTDATA.xlsx',1,['W',FirstRow]);
%Delay caused by navigation in and out of port [hours]
DelayNavigationPort = 2;

% Metocean data

%Read month and day, used for reading the appropriate weather data
Month = xlsread1('INPUTDATA.xlsx',1,['B',FirstRow,]);
Day = xlsread1('INPUTDATA.xlsx',1,['C',FirstRow]);

%Initialize counter for the number of Hs read
HsNumber = 0;

%Read all significant wave heights for the appropriate month and day in the
%time period 1992-2012 (3 hours interval)
for i = 1:size(Weather,1)
    %Can also define for the given time of the day
    if (Weather(i,2) == Month) && (Weather(i,3) == Day)
        %Note the number of Hs read
        HsNumber = HsNumber + 1;
        %Read the Hs from the weather data file
        Hsread(HsNumber) = abs(Weather(i,7));
    end
end

%Pick a random Hs from any of the same days from the 21 years of historical
%data
RandomDay = randi(HsNumber,1);

%If Hs is less than or equal to zero, then signal is either not read or
%misread, and another random day will be picked
while Hsread(RandomDay) <= 0
    RandomDay = randi(HsNumber,1);
end

%Set the appropriate significant wave height
Hs = Hsread(RandomDay);

```

### A.3 Calculations

```

%% Get distance from mapping, calculate sailing time and fuel consumption

%Calculating the distances between vessels and oil spill, and vessels and
port (in nautical miles)
for i=1:nVessels
    LocationVessel = [Latitude(i), Longitude(i)];
    %Distance of the great circle route
    %Distance from vessel location to oil spill
    DistOilSpillGC(i) = distance('gc',LocationVessel,LocationOilSpill);
    %Distance from vessel location to port
    DistPortGC(i) = distance('gc',LocationVessel,LocationPort);
    %Distance from port to oil spill
    DistPortOilGC(i) = distance('gc',LocationPort,LocationOilSpill);
    %Rhumb line: DistRL(i) =
distance('rh',LocationVessel,LocationOilSpill);
    %Distance in nautical miles
    NMDistOilSpillGC(i) = deg2nm(DistOilSpillGC(i));
    NMDistPortGC(i) = deg2nm(DistPortGC(i));
    NMDistPortOilGC(i) = deg2nm(DistPortOilGC(i));

    %Calculate sailing time in hours (affected by Hs)
    %Sailing time to oil spill (max speed)
    SailingTimeOilSpill(i) = ((NMDistOilSpillGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time to port (max speed)
    SailingTimePort(i) = ((NMDistPortGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time from port to oil spill (max speed)
    SailingTimePortOil(i) = ((NMDistPortOilGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time return trip from oil spill to port, in economical speed
    %after oil spill containment operation is complete
    SailingTimeReturn(i) =
((NMDistPortOilGC(i)/(SpeedEco(i)*0.5))*1.02^Hs);

    %Calculate fuel consumption for said trip
    %Weather affects sailing time, which again increases fuel consumption
    %Fuel consumption - Vessel location to oil spill
    FuelConsumptionSailToOilSpill(i) =
SailingTimeOilSpill(i)*FuelConMax(i);
    %Fuel consumption - Vessel location to port
    FuelConsumptionSailToPort(i) = SailingTimePort(i)*FuelConMax(i);
    %Fuel consumption - Port to oil spill
    FuelConsumptionPortToOilSpill(i) = SailingTimePortOil(i)*FuelConMax(i);
    %Fuel consumption - return trip in fuel save mode
    FuelConsumptionReturn(i) = SailingTimeReturn(i)*FuelConEco(i);

    %Time in oil spill operation [hours], very rough estimate
    TimeOperation = 10;

    %Time to refuel in port
    TimeRefuelPort(i) = ((FuelCapacity(i) - FuelLev(i) +
FuelConsumptionSailToPort(i))/RefuelingRatePort) + DelayNavigationPort;
end

```

```

%% Fuel Feasibility Criteria

%REFUEL AT OIL SPILL SITE
%Refuel at oil spill site, send other ship to pick up fuel
%Safety factor of 20 % included
RefuelPort = zeros(nVessels,nVessels);
for i=1:nVessels
    for j=1:nVessels
        if i ~= j
            %Fuel consumed while waiting for other vessel to arrive for
refueling
                %That is, fuel consumed while sailing out to oil spill, and
                %fuel consumed in operation at oil spill site while waiting for
                %the fuel support vessel to arrive
                RefuelPort(i,j) = FuelConsumptionSailToOilSpill(i) +
((SailingTimePort(j) + TimeRefuelPort(j) + SailingTimePortOil(j) -
SailingTimeOilSpill(i))*FuelConOperation(i));
                if RefuelPort(i,j)*1.2 <= FuelLev(i)
                    Refuel(i,j) = j;
                else
                    Refuel(i,j) = 0;
                end
            end
        end
    end
end

%NO REFUELING
for i=1:nVessels
    FuelConsumptionResponse(i) = FuelConsumptionSailToOilSpill(i) +
FuelConsumptionReturn(i) + (TimeOperation*FuelConOperation(i));
    if FuelConsumptionResponse(i)*1.2 <= FuelLev(i)
        FuelCriteria(i) = 1;
    else
        FuelCriteria(i) = 0;
    end
end
end

```



## A.4 Determining the Response

```

%% Decide how to respond to the emergency and which vessels to send

%This loop goes through a set of logical decisions and determines
%the fastest feasible response for each vessel
for i=1:nVessels
    %Feasibility of vessel (0 if not feasible)
    Vessels(i)=OpStatus(i)*CargoStatus(i)*Fleet(i);
    %Check if vessel is feasible for direct response
    if Vessels(i) > 0 && FuelCriteria(i) > 0 && EmergencyEquipment(i) ~= 0
        %If vessel is in port, add 4 hours delay
        if OpStatus(i) == 1
            Delay = 4;
        %If vessel is in transit towards platform
        elseif OpStatus(i) == 2
            Delay = 0;
        %if vessel is sailing return trip, add 0.2 h. Should also
        %contribute to fuel consumption?
        elseif OpStatus(i) == 3
            Delay = 0.2;
        end
        ResponseDirect(i) = SailingTimeOilSpill(i) + Delay
        ResponseRefuel(i) = 0;
        ResponseByPort(i) = 0;
        %If the ship's current fuel level is a limiting factor, then
        %check the option of being refueled by another ship while doing oil
        %spill operations
        elseif Vessels(i) >0 && max(Refuel(i,:)) >0 && EmergencyEquipment(i)~=0
            %If vessel is in port, add 4 hours delay
            if OpStatus(i) == 1
                Delay = 4;
            %If vessel is in transit towards platform
            elseif OpStatus(i) == 2
                Delay = 0;
            %if vessel is sailing return trip, add 0.2 h. Should also
            %contribute to fuel consumption?
            elseif OpStatus(i) == 3
                Delay = 0.2;
            end
            ResponseDirect(i) = 0;
            ResponseRefuel(i) = SailingTimeOilSpill(i) + Delay;
            ResponseByPort(i) = 0;
            %If the current fuel level is so low that the ship are both unable to
            %perform oil spill operations independently, and unable to wait for
            %another ship to refuel, then it must travel back to port and refuel.
            %This option is also chosen if the vessel does not have the
            %necessary equipment onboard to deal with the emergency, and must thus
            %pick this up at port.
        else
            %If vessel is in port, add 4 hours delay
            if OpStatus(i) == 1
                Delay = TimeRefuelPort(i);
                ResponseByPort(i) = SailingTimePortOil(i);
            %If vessel is in transit towards platform, turn around
            elseif OpStatus(i) == 2
                Delay = 0.2;
                ResponseByPort(i) = SailingTimePort(i) + TimeRefuelPort(i) +
                SailingTimePortOil(i);
            end
        end
    end
end

```

```

%if vessel is sailing return trip
    elseif OpStatus(i) == 3
        Delay = 0;
        ResponseByPort(i) = SailingTimePort(i) + TimeRefuelPort(i) +
            SailingTimePortOil(i);
    end
    SailingTimeOilSpill(i) = ResponseByPort(i);

    ResponseDirect(i) = 0;
    ResponseRefuel(i) = 0;
    ResponseByPort(i) = SailingTimeOilSpill(i) + Delay;
end
ResponseTime(i) = SailingTimeOilSpill(i) + Delay;
end

%Choose the minimum response time of all vessels as the solution,
%and print to command window the time and vessel chosen

[MinResponseTime, ChosenVessel] = min(ResponseTime(:))

%Initialize variable
RefueledByVessel = 0;
RefuelsAtPort = 0;

%Checking if the solution includes refueling by other vessel, and if so,
%which vessel to set to this task
for i=1:nVessels
    if MinResponseTime == ResponseRefuel(i)
        %Initialize the response time matrix of the fuel support vessels
        MinFuelSupportResponse = zeros(1,nVessels);
        %Checking the response time for the fuel support vessels
        for j=1:nVessels
            if Refuel(i,j) > 0
                MinFuelSupportResponse(j) = SailingTimePort(j) +
                    TimeRefuelPort(j) + SailingTimePortOil(j);
            end
        end
        %The vessel responding to the oil spill can not be chosen to act
        %as fuel support for itself, and is thus set with a response time
        %ten times that of the maximum response time for all fuel
        %supporting vessels
        for j=1:nVessels
            if MinFuelSupportResponse(j) == 0
                MinFuelSupportResponse(j) =
                    max(MinFuelSupportResponse(:))*10;
            end
        end
        %The vessel that can arrive first with extra fuel is chosen
        [MinFuelSupportResponseTime, FuelSupportVessel] =
            min(MinFuelSupportResponse(:));
        RefueledByVessel = FuelSupportVessel

    elseif MinResponseTime == ResponseDirect(i)
        disp('No refueling needed')
    elseif MinResponseTime == ResponseByPort(i)
        disp('Refuels at port')
        RefuelsAtPort = 1;
    end
end
end

```

## A.5 Plot the Solution

```

%Plot the chosen vessel and its route to the oil spill
figuremap = figure('color','w');
ha = axesm('mapproj','mercator','maplatlim',[70 78],'maplonlim',[50 90]);
axis off, gridm off, framem on;
mlabel on, plabel on;

>Loading coastline data
load coast
hg = geoshow(lat, long,'displaytype','line','color','b');
%Color land areas green
geoshow('landareas.shp','FaceColor',[0.15 0.5 0.15])

%Mark the locations of interest
for i=1:nVessels
    if i == ChosenVessel
        geoshow(Latitude(i),Longitude(i),'Marker','o','Color','red');
        %Label the location of the chosen vessel
        textm(Latitude(ChosenVessel)-0.3,Longitude(ChosenVessel)-3,
            'Chosen Vessel');
    elseif i == RefueledByVessel
        geoshow(Latitude(i),Longitude(i),'Marker','o','Color','green');
        %Label the location of the fuel support vessel
        textm(Latitude(i)-0.3,Longitude(i)-5, 'Fuel Support Vessel');
        %Plot the great circle route for the vessel chosen for fuel support
        GreatCirc =
            track2('gc',Latitude(FuelSupportVessel),Longitude(FuelSupportVessel),
                LocationPort(1),LocationPort(2));

        geoshow(GreatCirc(:,1),GreatCirc(:,2),'DisplayType','line','Color'
            ,'green','linestyle','-');
        GreatCirc =
            track2('gc',LocationPort(1),LocationPort(2),LocationOilSpill(1),Lo
                cationOilSpill(2));

        geoshow(GreatCirc(:,1),GreatCirc(:,2),'DisplayType','line','Color'
            ,'green','linestyle','-');
    else
        geoshow(Latitude(i),Longitude(i),'Marker','x','Color','black');
        %Label the location of the remaining vessels
        textm(Latitude(i)-0.3,Longitude(i)-3.5, 'Other Vessel');
    end
end

geoshow(LocationOilSpill(1),LocationOilSpill(2),'Marker','o','Color','black
');
%Label the location of the oil spill
textm(LocationOilSpill(1)+0.3,LocationOilSpill(2)-2, 'Oil Spill');

geoshow(LocationOilSpill(1),LocationOilSpill(2),'Marker','.', 'Color','yellow
');
%Label the location of the port
textm(LocationPort(1)-0.25,LocationPort(2)-0.2, 'Port');

```

```
%Create the great circle route from the chosen vessel to the oil spill
%If the vessel requires a stop at port
if RefuelsAtPort == 1
    GreatCirc =
        track2('gc',Latitude(ChosenVessel),Longitude(ChosenVessel),LocationPort
            (1),LocationPort(2));

    geoshow(GreatCirc(:,1),GreatCirc(:,2),'DisplayType','line','Color','red
        ','linestyle','-');
    GreatCirc =
        track2('gc',LocationPort(1),LocationPort(2),LocationOilSpill(1),Locatio
            nOilSpill(2));

    geoshow(GreatCirc(:,1),GreatCirc(:,2),'DisplayType','line','Color','red
        ','linestyle','-');
%If sailing directly to the oil spill
else
    GreatCirc =
        track2('gc',Latitude(ChosenVessel),Longitude(ChosenVessel),LocationOils
            pill(1),LocationOilSpill(2));

    geoshow(GreatCirc(:,1),GreatCirc(:,2),'DisplayType','line','Color','red
        ','linestyle','-');
end

%Closes excel, which have been held open through the entire simulation
Excel.ActiveWorkbook.Save;
Excel.Quit
Excel.delete
clear Excel
```

## Appendix B: xlsread1.m

The function *xlsread1.m* was created and published by user Antonio of the Federal University at Campina Grande on the MathWorks official forums in December 2008.

```
function [data, text, rawData,
customOutput]=xlsread1(file,sheet,range,mode,customFun)
% XLSREAD Get data and text from a spreadsheet in an Excel workbook.
% [NUMERIC,TXT,RAW]=XLSREAD(FILE) reads the data specified in the Excel
% file, FILE. The numeric cells in FILE are returned in NUMERIC, the text
% cells in FILE are returned in TXT, while the raw, unprocessed cell
% content is returned in RAW.
%
% [NUMERIC,TXT,RAW]=XLSREAD(FILE,SHEET,RANGE) reads the data specified
% in RANGE from the worksheet SHEET, in the Excel file specified in FILE.
% It is possible to select the range of data interactively (see Examples
% below). Please note that the full functionality of XLSREAD depends on
% the ability to start Excel as a COM server from MATLAB.
%
% [NUMERIC,TXT,RAW]=XLSREAD(FILE,SHEET,RANGE,'basic') reads an XLS file
as
% above, using basic input mode. This is the mode used on UNIX platforms
% as well as on Windows when Excel is not available as a COM server.
% In this mode, XLSREAD does not use Excel as a COM server, which limits
% import ability. Without Excel as a COM server, RANGE will be ignored
% and, consequently, the whole active range of a sheet will be imported.
% Also, in basic mode, SHEET is case-sensitive and must be a string.
%
% [NUMERIC,TXT,RAW]=XLSREAD(FILE,SHEET,RANGE,'',CUSTOMFUN)
% [NUMERIC,TXT,RAW,CUSTOMOUTPUT]=XLSREAD(FILE,SHEET,RANGE,'',CUSTOMFUN)
% When the Excel COM server is used, allows passing in a handle to a
% custom function. This function will be called just before retrieving
% the actual data from Excel. It must take an Excel Range object (e.g. of
% type 'Interface.Microsoft_Excel_5.0_Object_Library.Range') as input,
% and return one as output. Optionally, this custom function may return
% a second output argument, which will be returned from XLSREAD as the
% fourth output argument, CUSTOMOUTPUT. For details of what is possible
% using the EXCEL COM interface, please refer to Microsoft documentation.
%
% INPUT PARAMETERS:
% FILE: string defining the file to read from. Default directory is pwd.
%       Default extension is 'xls'.
% SHEET: string defining worksheet name in workbook FILE.
%        double scalar defining worksheet index in workbook FILE. See
%        NOTE 1.
% RANGE: string defining the data range in a worksheet. See NOTE 2.
% MODE: string enforcing basic import mode. Valid value = 'basic'. This
%       is the mode always used when COM is not available (e.g. on Unix).
%
% RETURN PARAMETERS:
% NUMERIC = n x m array of type double.
% TXT = r x s cell string array containing text cells in RANGE.
% RAW = v x w cell array containing unprocessed numeric and text data.
% Both NUMERIC and TXT are subsets of RAW.
%
%
```

## EXAMPLES:

```

% 1. Default operation:
%   NUMERIC = xlsread(FILE);
%   [NUMERIC,TXT]=xlsread(FILE);
%   [NUMERIC,TXT,RAW]=xlsread(FILE);
%
% 2. Get data from the default region:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet')
%
% 3. Get data from the used area in a sheet other than the first sheet:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet','sheet2')
%
% 4. Get data from a named sheet:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet','NBData')
%
% 5. Get data from a specified region in a sheet other than the first
%   sheet:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet','sheet2','a2:j5')
%
% 6. Get data from a specified region in a named sheet:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet','NBData','a2:j5')
%
% 7. Get data from a region in a sheet specified by index:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet',2,'a2:j5')
%
% 8. Interactive region selection:
%   NUMERIC = xlsread('c:\matlab\work\myspreadsheet',-1);
%   You have to select the active region and the active sheet in the
%   EXCEL window that will come into focus. Click OK in the Data
%   Selection Dialog when you have finished selecting the active region.
%
% 9. Using the custom function:
%   [NUMERIC,TXT,RAW,CUSTOMOUTPUT] = xlsread('equity.xls', ...,
@MyCustomFun)
%   Where the CustomFun is defined as:
%
%   function [DataRange, customOutput] = MyCustomFun(DataRange)
%       DataRange.NumberFormat = 'Date';
%       customOutput = 'Anything I want';
%
%   This will convert to dates all cells where that is possible.
%
% NOTE 1: The first worksheet of the workbook is the default sheet. If
% SHEET is -1, Excel comes to the foreground to enable interactive
% selection (optional). In interactive mode, a dialogue will prompt
% you to click the OK button in that dialogue to continue in
MATLAB.
%   (Only supported when Excel COM server is available.)
% NOTE 2: The regular form is: 'D2:F3' to select rectangular region D2:F3
% in a worksheet. RANGE is not case sensitive and uses Excel A1
% notation (see Excel Help). (Only supported when Excel COM server
% is available.)
% NOTE 3: Excel formats other than the default can also be read.
%   (Only supported when Excel COM server is available.)
%
% See also XLSWRITE, CSVREAD, CSVWRITE, DLMREAD, DLMWRITE, TEXTSCAN.

```

```

% Copyright 1984-2007 The MathWorks, Inc.
% $Revision: 1.23.4.24 $ $Date: 2007/12/06 13:30:15 $

```

```

=====
===

```

```
Excel = evalin('base','Excel'); % added command (Brandao 12/09/2008)
% initialise variables
data = [];
text = {};
rowData = {};

Sheet1 = 1;
if nargin < 2
    sheet = Sheet1;
    range = '';
elseif nargin < 3
    range = '';
end

% handle input values
if nargin < 1 || isempty(file)
    error('MATLAB:xlsread:FileName','Filename must be specified.');
```

```
end

if ~ischar(file)
    error('MATLAB:xlsread:InputClass','Filename must be a string.');
```

```
end

if nargin > 1
    % Verify class of sheet parameter
    if ~ischar(sheet) && ...
        ~(isnumeric(sheet) && length(sheet)==1 && ...
            floor(sheet)==sheet && sheet >= -1)
        error('MATLAB:xlsread:InputClass',...
            'Sheet argument must a string or an integer.');
```

```
end

if isequal(sheet,-1)
    range = ''; % user requests interactive range selection.
elseif ischar(sheet)
    if ~isempty(sheet)
        % Parse sheet and range strings
        if isempty(strfind(sheet,':'))
            else
                range = sheet; % only range was specified.
                sheet = Sheet1;% Use default sheet.
            end
        else
            sheet = Sheet1; % set sheet to default sheet.
        end
    end
end

if nargin > 2
    % verify class of range parameter
    if ~ischar(range)
        error('MATLAB:xlsread:InputClass',...
            'Range argument must a string. See HELP XLSREAD.');
```

```
end

end

if nargin >= 4
    % verify class of mode parameter
    if ~isempty(mode) && ~(strcmpi(mode,'basic'))
        warning('MATLAB:xlsread:InputClass',...
            'Mode argument must be basic.');
```

```

        'Import mode string is invalid. XLSREAD resets mode to
normal.');
```

mode = '';

```

    end
else
    mode = '';
end

custom = false;
if nargin >= 5
    if strcmpi(mode,'basic') || ~ispc
        warning('MATLAB:xlsread:Incompatible',...
            ['Custom functions cannot be used in basic mode or on non-Windows
platforms.\n'...
            'The custom function argument will be ignored.'])
    elseif ~isa(customFun,'function_handle')
        warning('MATLAB:xlsread:NotHandle', ...
            'The fifth argument to XLSREAD must be a function
handle.');
```

else

```

        custom = true;
    end
end

%=====
%====
% block command crossed out (Brandao 12/09/2008)
% handle requested Excel workbook filename
% try
%     file = validpath(file, '.xls');
% catch exception
%     err = MException('MATLAB:xlsread:FileNotFound','XLSREAD unable to
open file %s.\n%s',...
%                                     file,exception.message);
%     throw(err);
% end
%=====
%====
% select import mode from either normal or basic mode.
if strcmpi(mode,'basic') || ~ispc
    warning('MATLAB:xlsread:Mode',...
        ['XLSREAD has limited import functionality on non-Windows
platforms\n'...
        'or in basic mode. Refer to HELP XLSREAD for more
information.']);
    try
        if nargin > 2
            [data,text,rawData] = xlsreadold(file,sheet);
        else
            [data,text] = xlsreadold(file,sheet);
        end
    catch exception
        if isempty(exception.identifier)
            exception = MException('MATLAB:xlsreadold:FormatError','%s',
exception.message);
        end
        throw(exception);
    end
    return;
% else % block command crossed out (Brandao 12/09/2008)
%     % Attempt to start Excel as ActiveX server process.
```



```

%      try
%      Excel = actxserver('excel.application');
%      catch exc1
%      % revert to old XLSREAD that uses BIFFREAD
%      warning('MATLAB:xlsread:ActiveX',...
%      ['Could not start Excel server for import. '...
%      'Refer to documentation.']);
%      try
%      if nargin > 2
%      [data,text,rawData] = xlsreadold(file, sheet);
%      else
%      [data,text] = xlsreadold(file, sheet);
%      end
%      catch exc2
%      message=sprintf('%s\n%s', exc1.message, exc2.message);
%      if isempty(exc2.identifier)
%      exception = MException('MATLAB:xlsreadold:FormatError',
message);
%      else
%      exception = MException(exc2.identifier, message);
%      end
%      throw(exception);
%      end
%      return;
%      end
end
=====
====
try
% open workbook
Excel.DisplayAlerts = 0;

%Workaround for G313142. For certain files, unless a workbook is
%opened prior to opening the file, various COM calls return an error:
%0x800a9c64. The line below works around this flaw. Since we have
%seen only one example of such a file, we have decided not to incur the
%time penalty involved here.
%      aTemp = Excel.workbooks.Add(); aTemp.Close();

%      try % block command crossed out (Brandao 12/09/2008)
%      ExcelWorkbook = Excel.workbooks.Open(file,0,true);
%      catch exception %#OK
%      %do not pollute lasterror state
%      end

%      % block command crossed out (Brandao 12/09/2008)
%      format = ExcelWorkbook.FileFormat;
%      if strcmpi(format, 'xlCurrentPlatformText') == 1
%      error('MATLAB:xlsread:FileFormat', 'File %s not in Microsoft
Excel Format.', file);
%      end

if nargin >= 2
% User specified at least a worksheet or interactive range
selection.
if ~isequal(sheet,-1)
% Activate indicated worksheet.
activate_sheet(Excel, sheet);

```

```

try % importing a data range.
    if ~isempty(range)
        % The range is specified.
        Select(Range(Excel, sprintf('%s', range)));
        DataRange = get(Excel, 'Selection');
    else
        % Only the worksheet is specified.
        % Activate upper left cell on sheet.
        Activate(Range(Excel, 'A1'));

        % Select range of occupied cells in active sheet.
        DataRange = Excel.ActiveSheet.UsedRange;
    end
catch % data range error.
    error('MATLAB:xlsread:RangeSelection',...
        'Data range is invalid.');
```

end

```

else
    % User requests interactive range selection.
    % Set focus to first sheet in Excel workbook.
    activate_sheet(Excel, Sheet1);

    % Make Excel interface the active window.
    set(Excel, 'Visible', true);

    % bring up message box to prompt user.
    uiwait(warndlg({'Select data region in Excel worksheet.';...
        'Click OK to continue in MATLAB'},...
        'Data Selection Dialogue', 'modal'));
    DataRange = get(Excel, 'Selection');
    set(Excel, 'Visible', false); % remove Excel interface from
desktop
    end
else
    % No sheet or range or interactive range selection.
    % Activate default worksheet.
    activate_sheet(Excel, Sheet1);

    % Select range of occupied cells in active sheet.
    DataRange = Excel.ActiveSheet.UsedRange;
end

%Call the custom function if it was given. Provide customOutput if it
%is possible.
if custom
    if nargin(customFun) < 2
        DataRange = customFun(DataRange);
        customOutput = {};
    else
        [DataRange, customOutput] = customFun(DataRange);
    end
end

% get the values in the used regions on the worksheet.
rowData = DataRange.Value;
% parse data into numeric and string arrays
[data, text] = parse_data(rowData);

catch exception
```

```

% try % block command crossed out (Brandao 12/09/2008)
% ExcelWorkbook.Close(false); % close workbook without saving any
changes
% catch exc2 %#OK
% %Do not pollute lasterror state
% end
% rethrow(exception); % rethrow original error
end

% try % block command crossed out (Brandao 12/09/2008)
% ExcelWorkbook.Close(false); % close workbook without saving any
changes
% %This call could fail if the file is "locked". This is the same
% %message you would get if you opened the file in Excel, and then
tried
% %to close the workbook (NOT the application).
% Excel.Quit;
% catch exception
% warning(exception.identifier, '%s', exception.message);
% Excel.Quit;
% end

%-----
function [numericArray,textArray] = parse_data(data)
% PARSE_DATA parse data from raw cell array into a numeric array and a text
% cell array.
% [numericArray,textArray] = parse_data(data)
% Input:
% data: cell array containing data from spreadsheet
% Return:
% numericArray: double array containing numbers from spreadsheet
% textArray: cell string array containing text from spreadsheet
%=====

% ensure data is in cell array
if ischar(data)
    data = cellstr(data);
elseif isnumeric(data) || islogical(data)
    data = num2cell(data);
end

% Check if raw data is empty
if isempty(data)
    % Abort when all data cells are empty.
    textArray = {};
    numericArray = [];
    return
else
    % Trim empty leading and trailing rows
    % find empty cells
    emptycells = cellfun('isempty',data);
    nrows = size(emptycells,1);
    firstrow = 1;
    % find last of leading empty rows
    while (firstrow<=nrows && all(emptycells(firstrow,:)))
        firstrow = firstrow+1;
    end
    % remove leading empty rows
    data = data(firstrow:end,:);

```

```

% find start of trailing empty rows
nrows = size(emptycells,1);
lastrow = nrows;
while (lastrow>0 && all(emptycells(lastrow,:)))
    lastrow = lastrow-1;
end
% remove trailing empty rows
data = data(1:lastrow,:);

% find start of trailing NaN rows
warning('off', 'MATLAB:nonIntegerTruncatedInConversionToChar');
while (lastrow>0 && ~(any(cellfun('islogical', data(lastrow,:)))) &&
...
    all(isnan([data{lastrow,:}])))
    lastrow = lastrow-1;
end
warning('on', 'MATLAB:nonIntegerTruncatedInConversionToChar');
% remove trailing NaN rows
data=data(1:lastrow,:);

[n,m] = size(data);
textArray = cell(size(data));
textArray(:) = {' '};
end

vIsNaN = false(n,m);

% find non-numeric entries in data cell array
vIsText = cellfun('isclass',data,'char');
vIsNaN =
cellfun('isempty',data)|strcmpi(data,'nan')|cellfun('isclass',data,'char');

% place text cells in text array
if any(vIsText(:))
    textArray(vIsText) = data(vIsText);
else
    textArray = {};
end
% Excel returns COM errors when it has a #N/A field.
textArray = strrep(textArray,'ActiveX_VT_ERROR: ','#N/A');

% place NaN in empty numeric cells
if any(vIsNaN(:))
    data(vIsNaN)={NaN};
end

% extract numeric data
data = reshape(data,n,m);
rows = size(data,1);
m = cell(rows,1);
% Concatenate each row first
for n=1:rows
    m{n} = cat(2,data{n,:});
end
% Now concatenate the single column of cells into a matrix
numericArray = cat(1,m{:});

% trim all-NaN leading rows and columns from numeric array

```

```

% trim all-empty trailing rows and columns from text arrays
[numericalArray,textArray]=trim_arrays(numericalArray,textArray);

% ensure numericalArray is 0x0 empty.
if isempty(numericalArray)
    numericalArray = [];
end

%-----
function activate_sheet(Excel,Sheet)
% Activate specified worksheet in workbook.

% Initialise worksheet object
WorkSheets = Excel.sheets;

% Get name of specified worksheet from workbook
try
    TargetSheet = get(WorkSheets,'item',Sheet);
catch
    error('MATLAB:xlsread:WorksheetNotFound',...
        'Specified worksheet was not found.');
```

```

end

%Activate silently fails if the sheet is hidden
set(TargetSheet, 'Visible','xlSheetVisible');
% activate worksheet
Activate(TargetSheet);

%-----
function [matrixResult,cellResult,rowResult]=xlsreadold(filename, sheet)
% Basic import mode. Range specification not available.
% Interactive range selection not available.
% Read Excel file as binary image file
if nargin > 1
    if isequal(sheet,1) || isequal(sheet,-1)
        sheet = '';
    elseif ~ischar(sheet)
        error('MATLAB:xlsread:WorksheetNotFound',...
            'In basic mode, sheet argument must be a string.');
```

```

    end
end
% read XLS file
biffvector = biffread(filename);

% get sheet names
[data, names] = biffparse(biffvector);

% if the names array is empty, this is an old style biff record with
% no sheet name. Just return data and empty text cell array.
if isempty(names)
    matrixResult = data;
    cellResult = cell(names);
    if nargin > 2
        rowResult = num2cell(data);
    end
    return;
end

if nargin == 1 || isempty(sheet)

```

```

    % just get the first sheet
    [n, s] = biffparse(biffvector, names{1});
else
    % try to read this sheet
    try
        [n, s] = biffparse(biffvector, sheet);
    catch
        error('MATLAB:xlsread:WorksheetNotFound',...
            'Specified worksheet was not found.');
```

end

```

end

% trim trailing empty text cells and NaN matrix elements
[matrixResult, cellResult] = trim_arrays(n,s);
% replace empty text cells with char([]).
cellResult(cellfun('isempty',cellResult))={' '};

if nargin > 2
    % create raw data return
    if isempty(s)
        rawResult = num2cell(n);
    else
        rawResult = cell(max(size(n),size(s)));
        rawResult(1:size(n,1),1:size(n,2)) = num2cell(n);
        for i = 1:size(s,1)
            for j = 1:size(s,2)
                if (~isempty(s{i,j}) && (i > size(n,1) || j > size(n,2) ||
isnan(n(i,j))))
                    rawResult(i,j) = s(i,j);
                end
            end
        end
    end
end
% trim all-empty-string leading rows from raw array
while size(rawResult,1)>1 && all(cellfun('isempty',rawResult(1,:)))
    rawResult = rawResult(2:end,:);
end
% trim all-empty-string leading columns from raw array
while size(rawResult,2)>1 && all(cellfun('isempty',rawResult(:,1)))
    rawResult = rawResult(:,2:end);
end
% replace empty raw data with NaN, to comply with specification
rawResult(cellfun('isempty',rawResult))={NaN};
end

%-----
function [numericArray,textArray] = trim_arrays(numericArray,textArray)
% trim leading rows or cols
% if the string result has dimensions corresponding to a column or row of
% zeros in the matrix result, trim the zeros.
if ~isempty(numericArray) && ~isempty(textArray)
    [mn, nn] = size(numericArray);
    [ms, ns] = size(textArray);

    if ms == mn
        % trim leading column(textArray) from numeric data
        firstcol = 1;
        while (firstcol<=nn && all(isnan(numericArray(:,firstcol))))
            firstcol = firstcol+1;
        end
    end
end

```

```

        numericArray=numericArray(:,firstcolm:end);
    end

    if ns == nn
        % trim leading NaN row(s) from numeric data
        firstrow = 1;
        while (firstrow<=mn && all(isnan(numericArray(firstrow,:))))
            firstrow = firstrow+1;
        end
        numericArray=numericArray(firstrow:end,:);

        % trim leading empty rows(s) from text data
        firstrow = 1;
        while (firstrow<=ms &&
all(cellfun('isempty',textArray(firstrow,:))))
            firstrow = firstrow+1;
        end
        textArray=textArray(firstrow:end,:);
    end

    % trim all-empty-string trailing rows from text array
    lastrow = size(textArray,1);
    while (lastrow>0 && all(cellfun('isempty',textArray(lastrow,:))))
        lastrow = lastrow-1;
    end
    textArray=textArray(1:lastrow,:);

    % trim all-empty-string trailing columns from text array
    lastcol = size(textArray,2);
    while (lastcol>0 && all(cellfun('isempty',textArray(:,lastcol))))
        lastcol = lastcol-1;
    end
    textArray=textArray(:,1:lastcol);

    % trim all-NaN trailing rows from numeric array
    lastrow = size(numericArray,1);
    while (lastrow>0 && all(isnan(numericArray(lastrow,:))))
        lastrow=lastrow-1;
    end
    numericArray=numericArray(1:lastrow,:);

    % trim all-NaN trailing columns from numeric array
    lastcol = size(numericArray,2);
    while (lastcol>0 && all(isnan(numericArray(:,lastcol))))
        lastcol=lastcol-1;
    end
    numericArray=numericArray(:,1:lastcol);
end

```





## Appendix C: Input Data Sheet

The input data is in the Excel sheet given column by column, i.e. the blue heading continues along the same row.

Month	Day	Vessel nr	Max Speed [knots]	Service Speed [knots]
	2	12	1	14.6
			2	14.6
			3	14.6

Economic Speed [knots]	Fuel Consumption in Max Speed [tonne/day]	Fuel Consumption in Service Speed [tonne/day]	Fuel Consumption in Economic Speed [tonne/day]	Fuel Consumption in Operation [tonne/day]
10	19.5	13.6	9.8	6
10	19.5	13.6	9.8	6
10	19.5	13.6	9.8	6

Fuel Capacity [tonne]	Current Fuel Level [tonne]	Operational Status (0,1,2 or 3)	Cargo Status (1 or 0)	Emergency Equipment Availability (1 or 0)
300	134	1	1	1
300	97	3	1	1
300	281	2	1	1

Latitude of vessel	Longitude of vessel	Latitude of Port	Longitude of Port	Latitude of Oil Spill
72.48	72.86	72.48	72.86	76
74.67	72.18			
73.92	71.67			

Longitude of Oil Spill	Refueling Rate of the Port [tonne/hour]
75	200



## Appendix D: Case Study Simulation Model

```

%% Initialization

%Clear out all data and figures for a clean run
clc
clear all
close all

%Initializes the function xlsread1.m which keeps the excel sheet open
%throughout the simulation, instead of opening and closing excel each
%time a parameter is read. This caused tremendous saving in the program
%run time; After implementing xlsread1.m, the simulation run time was
%reduced to about 22 % of the original simulation time.

%Requires xlsread1.m in the same folder as this program

Excel = actxserver ('Excel.Application');
%NB!
%REMEMBER TO DEFINE THE PATH OF THE INPUT DATA FILE BELOW
File='C:\Users\David\Documents\NTNU\MASTER\MATLAB\INPUTDATA.xlsx';
if ~exist(File,'file')
    ExcelWorkbook = Excel.Workbooks.Add;
    ExcelWorkbook.SaveAs(File,1);
    ExcelWorkbook.Close(false);
end
Excel.Workbooks.Open(File);

%% Input Data that is valid for each simulation
InitializeRow = 3;
FirstRow = num2str(InitializeRow);
%Fleet Characteristics
%Vessel data that will serve as input
Counter = 1;
FleetRow = FirstRow;
while isempty(xlsread1('INPUTDATA.xlsx',1,['D',FleetRow])) == 0
Fleet(Counter) = xlsread1('INPUTDATA.xlsx',1,['D',FleetRow]);
Counter = Counter + 1;
FleetRow = str2num(FleetRow);
FleetRow = FleetRow + 1;
FleetRow = num2str(FleetRow);
end
EndRow = num2str(str2num(FleetRow) - 1);
%Number of Vessels
nVessels = length(Fleet);
%Max speed [knots]
SpeedMax = xlsread1('INPUTDATA.xlsx',1,['E',FirstRow,':','E',EndRow]);
%Service speed [knots]
SpeedService = xlsread1('INPUTDATA.xlsx',1,['F',FirstRow,':','F',EndRow]);
%Economical speed [knots]
SpeedEco = xlsread1('INPUTDATA.xlsx',1,['G',FirstRow,':','G',EndRow]);
%Fuel consumption in max speed [tonne/hour]
FuelConMax = xlsread1('INPUTDATA.xlsx',1,['H',FirstRow,':','H',EndRow])/24;
%Fuel consumption in service speed [tonne/hour]
FuelConService =
xlsread1('INPUTDATA.xlsx',1,['I',FirstRow,':','I',EndRow])/24;

```

```

%Fuel consumption in economical speed [tonne/hour]
FuelConEco = xlsread1('INPUTDATA.xlsx',1,['J',FirstRow,':','J',EndRow])/24;
%Fuel consumption in operation [tonne/hour]
FuelConOperation =
xlsread1('INPUTDATA.xlsx',1,['K',FirstRow,':','K',EndRow])/24;
%Fuel capacity of the fleet [tonne]
FuelCapacity = xlsread1('INPUTDATA.xlsx',1,['L',FirstRow,':','L',EndRow]);
%Emergency Equipment Status (0 if unequipped, 1 if equipment is available)
EmergencyEquipment =
xlsread1('INPUTDATA.xlsx',1,['P',FirstRow,':','P',EndRow]);

%Location of port in latitudes and longitudes
LocationPort = [xlsread1('INPUTDATA.xlsx',1,['S',FirstRow]),
xlsread1('INPUTDATA.xlsx',1,['T',FirstRow])];
%Location of oil spill in latitudes and longitudes
LocationOilSpill = [xlsread1('INPUTDATA.xlsx',1,['U',FirstRow]),
xlsread1('INPUTDATA.xlsx',1,['V',FirstRow])];

%Refueling rate of the port [tonne/hour]
RefuelingRatePort = 200;
%Delay caused by navigation in and out of port [hours]
DelayNavigationPort = 2;

%Read csv datafile into a matrix
Weather = csvread('WeatherData.csv',0,0);

%Choose the number of simulations to repeat
NumberOfSimulations = 10000;
for simulations = 1:NumberOfSimulations
%% Input Data That Changes For Each Simulation

for i=1:nVessels
    %Current fuel level in [tonne]
    FuelLev(i) = (FuelCapacity(i)-50).*rand(1,1) + 50;

    %Operational status
    %0 = unable to abort current operation, 1 = in port, 2 = in
    %transit going towards platform (or at platform), 3 = return trip
    RndOp = randi(100,1);
    if RndOp <= 2
        OpStatus(i) = 0;
    elseif RndOp > 2 && RndOp <= 51
        OpStatus(i) = 2;
    elseif RndOp > 51 && RndOp <= 100
        OpStatus(i) = 3;
    end

    %Cargo status
    %Does delivery of cargo take priority or conflict with response?
    %0 = unable to do oil spill, 1 able to
    %The probability of Cargo Status = 0 is one in a thousand
    RndCargo = randi(1000,1);
    if RndCargo == 1
        CargoStatus(i) = 0;
    else
        CargoStatus(i) = 1;
    end

%Location of vessels is varied between the coordinates of the port and

```

```

%the oil spill. The sailing route is split into three legs so that the
%assigned vessel positions are more realistic.
%The latitude is decided between the port and the oil spill
Latitude(i) = ([LocationOilSpill(1)] - [LocationPort(1)].*rand(1,1) +
[LocationPort(1)]);
%If the latitude of the vessel position is within the first third of
%the route (closest to the port), then the longitude can be given as
%the third of the route that is closest to the port longitudes.
if Latitude(i) <= ((2/3)*[LocationPort(1)] +
(1/3)*[LocationOilSpill(1)]);
    Longitude(i) = (((2/3)*[LocationPort(2)] +
(1/3)*[LocationOilSpill(2)]) - [LocationPort(2)].*rand(1,1) +
[LocationPort(2)]);
%If the latitude is within the last third of the route (closest to oil
%spill), then the longitude is given as the third that is nearest the
%oil spill longitudes
elseif Latitude(i) >= ((1/3)*[LocationPort(1)] +
(2/3)*[LocationOilSpill(1)])
    Longitude(i) = ([LocationOilSpill(2)]-((1/3)*[LocationPort(2)] +
(2/3)*[LocationOilSpill(2)]).*rand(1,1) + ((1/3)*[LocationPort(2)] +
(2/3)*[LocationOilSpill(2)]);
%If the latitude is withing the middle third of the route, then the
%longitude is chosen accordingly.
else
    Longitude(i) = (((1/3)*[LocationPort(2)] +
(2/3)*[LocationOilSpill(2)]) - ((2/3)*[LocationPort(1)] +
(1/3)*[LocationOilSpill(1)]).*rand(1,1) + ((2/3)*[LocationPort(1)] +
(1/3)*[LocationOilSpill(1)]);
end
%Operational Status is set to 1 (in port) if the vessel latitude is
%within a 0.05 decimal degrees proximity to the location of the port
if Latitude(i) <= ([LocationPort(1)] + 0.05)
    OpStatus(i) = 1;
end
end

% Metocean data
%Read month and day, used for reading the appropriate weather data
%The month is referred to numerically, by a number from 1 to 12,
%where 1 is January and 12 is December
Month = randi(12,1);
%The day ranges from the 1st to the 31st for the following months
if Month == 1 || Month == 3 || Month == 5 || Month == 7 || Month == 8 ||
Month == 10 || Month == 12
    Day = randi(31,1);
%The day ranges from the 1st to the 30th for the following months
elseif Month == 4 || Month == 6 || Month == 9 || Month == 11
    Day = randi(30,1);
%The day ranges from the 1st to the 28th for February
%(leap year is not included)
else
    Day = randi(28,1);
end

%Initialize counter for the number of Hs read
HsNumber = 0;

```

```

%Read all significant wave heights for the appropriate month and day in the
%time period 1992-2012 (3 hours interval)
for i = 1:size(Weather,1)
    if (Weather(i,2) == Month) && (Weather(i,3) == Day)
        %Note the number of Hs read
        HsNumber = HsNumber + 1;
        %Read the Hs from the weather data file
        Hsread(HsNumber) = abs(Weather(i,7));
    end
end

%Pick a random Hs from any of the same days from the 21 years of historical
%data
RandomDay = randi(HsNumber,1);

%If Hs is less than or equal to zero, then signal is either not read or
%misread, and another random day will be picked
while Hsread(RandomDay) <= 0
    RandomDay = randi(HsNumber,1);
end

%Set the appropriate significant wave height
Hs = Hsread(RandomDay);

%% Get distance from mapping, calculate sailing time and fuel consumption

%Calculating the distances between vessels and oil spill, and vessels and
port (in nautical miles)
for i=1:nVessels
    LocationVessel = [Latitude(i), Longitude(i)];
    %Distance of the great circle route
    %Distance from vessel location to oil spill
    DistOilSpillGC(i) = distance('gc',LocationVessel,LocationOilSpill);
    %Distance from vessel location to port
    DistPortGC(i) = distance('gc',LocationVessel,LocationPort);
    %Distance from port to oil spill
    DistPortOilGC(i) = distance('gc',LocationPort,LocationOilSpill);
    %Rhumb line: DistRL(i) =
distance('rh',LocationVessel,LocationOilSpill);
    %Distance in nautical miles
    NMDistOilSpillGC(i) = deg2nm(DistOilSpillGC(i));
    NMDistPortGC(i) = deg2nm(DistPortGC(i));
    NMDistPortOilGC(i) = deg2nm(DistPortOilGC(i));

    %Calculate sailing time in hours (affected by Hs)
    %Sailing time to oil spill (max speed)
    SailingTimeOilSpill(i) = ((NMDistOilSpillGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time to port (max speed)
    SailingTimePort(i) = ((NMDistPortGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time from port to oil spill (max speed)
    SailingTimePortOil(i) = ((NMDistPortOilGC(i)/SpeedMax(i))*1.02^Hs);
    %Sailing time return trip from oil spill to port, in economical mode
    %after oil spill containment operation is complete (economical speed)
    SailingTimeReturn(i) =
((NMDistPortOilGC(i)/(SpeedEco(i)*0.5))*1.02^Hs);

```

```

%Calculate fuel consumption for said trip
%Weather affects sailing time, which again increases fuel consumption
%Fuel consumption - Vessel location to oil spill
FuelConsumptionSailToOilSpill(i) =
SailingTimeOilSpill(i)*FuelConMax(i);
%Fuel consumption - Vessel location to port
FuelConsumptionSailToPort(i) = SailingTimePort(i)*FuelConMax(i);
%Fuel consumption - Port to oil spill
FuelConsumptionPortToOilSpill(i) = SailingTimePortOil(i)*FuelConMax(i);
%Fuel consumption - return trip in fuel save mode
FuelConsumptionReturn(i) = SailingTimeReturn(i)*FuelConEco(i);

%Time in oil spill operation [hours], very rough estimate
TimeOperation = 10;

%Time to refuel in port
TimeRefuelPort(i) = ((FuelCapacity(i) - FuelLev(i) +
FuelConsumptionSailToPort(i))/RefuelingRatePort) + DelayNavigationPort;
end

%% Fuel Criteria

%REFUEL AT OIL SPILL SITE
%Refuel at oil spill site, send other ship to pick up fuel
%Safety factor of 20 % included
RefuelPort = zeros(nVessels,nVessels);
for i=1:nVessels
    for j=1:nVessels
        if i ~= j
            %Fuel consumed while waiting for other vessel to arrive for
            refueling
            %That is, fuel consumed while sailing out to oil spill, and
            %fuel consumed in operation at oil spill site while waiting for
            %the refuel ship
            RefuelPort(i,j) = FuelConsumptionSailToOilSpill(i) +
            ((SailingTimePort(j) + TimeRefuelPort(j) + SailingTimePortOil(j) -
            SailingTimeOilSpill(i))*FuelConOperation(i));
            if RefuelPort(i,j)*1.2 <= FuelLev(i)
                Refuel(i,j) = j;
            else
                Refuel(i,j) = 0;
            end
        end
    end
end
end

%NO REFUELING
for i=1:nVessels
    FuelConsumptionResponse(i) = FuelConsumptionSailToOilSpill(i) +
    FuelConsumptionReturn(i) + (TimeOperation*FuelConOperation(i));
    if FuelConsumptionResponse(i)*1.2 <= FuelLev(i)
        FuelCriteria(i) = 1;
    else
        FuelCriteria(i) = 0;
    end
end
end

```

```

%% Decide how to respond to the emergency and which vessels to send

%This loop goes through a set of logical decisions and determines
%the fastest feasible response for each vessel
for i=1:nVessels
    %Feasibility of vessel (0 if not feasible)
    Vessels(i)=OpStatus(i)*CargoStatus(i)*Fleet(i);
    %Check if vessel is unable to respond
    if Vessels(i) == 0
        Delay = SailingTimePortOil(i)*10;
        ResponseDirect(i) = 0;
        ResponseRefuel(i) = 0;
        ResponseByPort(i) = 0;
        %Check if vessel is feasible for direct response
        elseif Vessels(i) > 0 && FuelCriteria(i) > 0 && EmergencyEquipment(i)
~= 0
            %If vessel is in port, add 4 [hours] delay
            if OpStatus(i) == 1
                Delay = 4;
            %If vessel is in transit towards platform
            elseif OpStatus(i) == 2
                Delay = 0;
            %if vessel is sailing return trip, add 0.2 [hours]
            elseif OpStatus(i) == 3
                Delay = 0.2;
            end
            ResponseDirect(i) = SailingTimeOilSpill(i) + Delay;
            ResponseRefuel(i) = 0;
            ResponseByPort(i) = 0;
            %If the ship's current fuel level is a limiting factor, then
            %check the option of being refueled by another ship while doing oil
            %spill operations
            elseif Vessels(i) > 0 && max(Refuel(i,:)) > 0 && EmergencyEquipment(i)
~= 0
                %If vessel is in port, add 4 [hours] delay
                if OpStatus(i) == 1
                    Delay = 4;
                %If vessel is in transit towards platform
                elseif OpStatus(i) == 2
                    Delay = 0;
                %if vessel is sailing return trip, add 0.2 [hours] Should also
                %contribute to fuel consumption?
                elseif OpStatus(i) == 3
                    Delay = 0.2;
                end
                ResponseDirect(i) = 0;
                ResponseRefuel(i) = SailingTimeOilSpill(i) + Delay;
                ResponseByPort(i) = 0;

```



```

%If the current fuel level is so low that the ship are both unable to
%perform oil spill operations independently, and unable to wait for
%another ship to refuel, then it must travel back to port and refuel.
%This option is also chosen if the vessel does not have the
%necessary equipment onboard to deal with the emergency, and must thus
%pick this up at port.
else
    %If vessel is in port, add 4 [hours] delay
    if OpStatus(i) == 1
        Delay = TimeRefuelPort(i);
        ResponseByPort(i) = SailingTimePortOil(i);
    %If vessel is in transit towards platform, turn around
    elseif OpStatus(i) == 2
        Delay = 0.2;
        ResponseByPort(i) = SailingTimePort(i) + TimeRefuelPort(i) +
SailingTimePortOil(i);
    %if vessel is sailing return trip
    elseif OpStatus(i) == 3
        Delay = 0;
        ResponseByPort(i) = SailingTimePort(i) + TimeRefuelPort(i) +
SailingTimePortOil(i);
    end
    SailingTimeOilSpill(i) = ResponseByPort(i);

    ResponseDirect(i) = 0;
    ResponseRefuel(i) = 0;
    ResponseByPort(i) = SailingTimeOilSpill(i) + Delay;
end
ResponseTime(i) = SailingTimeOilSpill(i) + Delay;
end

%Choose the minimum response time of all vessels as the solution,
%and print to command window the time and vessel chosen

[MinResponseTime, ChosenVessel] = min(ResponseTime(:));

%Register the result of this particular simulation to a variable containing
%the results of all simulations
AllChosenResponseTimes(simulations) = MinResponseTime;

end

disp('Simulations Complete')
disp('All simulated response times can be found in the variable
"AllChosenResponseTimes".')

%Closes excel, which have been held open through the entire simulation
Excel.ActiveWorkbook.Save;
Excel.Quit
Excel.delete
clear Excel

```



## Appendix E: Abbreviations

CSV	Comma Separated Values
HSE	Health, Safety & Environment
NCA	Norwegian Coastal Administration
NOFO	Norwegian Clean Seas Association for Operation Companies
OSCAR	Oil Spill Contingency and Response
PSA	Petroleum Safety Authority Norway
SAR	Search and Rescue
SF	Safety Factor