PROBABILITY OF SHIP COLLISION WITH OFFSHORE WIND FARMS IN THE SOUTHERN NORTH SEA

by

Alin Gavril Chițu

A thesis submitted to the Delft University of Technology in conformity with the requirements for the degree of

Master of Science

Delft, The Netherlands

July 2005

Approved by ____________ Prof. Dr. Roger M. Cooke ____________
Chairperson of Supervisory Committee

____________ Dr. Dorota Kurowicka ____________

____________ Dr. Ulrich Callies ____________

Delft University of Technology

Abstract

PROBABILITY OF SHIP COLLISION WITH OFFSHORE WIND FARMS IN THE SOUTHERN NORTH SEA

by Alin Gavril Chițu

Chairperson of the Supervisory Committee: Professor Roger M. Cooke
Department of Applications of Decision Theory

The Wadden Sea lies between a section of the coast of continental Europe and the North Sea. It extends from Den Helder in The Netherlands in the southwest, encloses the river estuaries of Germany at north, to Esbjerg in Denmark. The Wadden Sea is famous for its unique ecosystem. Following a trend that appeared all over the world in the last decade, namely to support renewable energy sources, the German authorities are planning to build offshore windmill farms in the German Bight. The general concern is that placing the windmill farms in this region will increase the risk of having an accident implicating one of the tankers from the two shipping routes that are crossing the area. This project describes the consequences of placing the windmill farms in the given area on the ecosystem of the region. The probability of collision of a drifting boat with a wind turbine was computed based on detailed reconstruction of environmental conditions. One important aspect is that wind conditions limit the time available for emergency towing operations to avert a collision of a vessel in distress with a wind turbine. Hence the model developed also includes towboats as a possible mitigation measure.

TABLE OF CONTENTS

LIST OF FIGURES

Figure number and caption *Page*

ACKNOWLEDGMENTS

I would first like to thank Prof. Roger Cooke and Prof. Arnold Heemink for giving me the opportunity to study at TUDelft. I have learned many great things during these two years, not only in the field of mathematics but also about research and life in general. I would like to acknowledge and present my sincere appreciation to all the teachers that coordinated our study here in Delft. The ones with the most important role in our formation are: Roger Cooke, Dorota Kurowicka, Thomas Mazzuchi, Arnold Heemink, Peter Wilders, Jan van Noortwijk, and Hans van der Weide.

I also want to thank Ulrich Callies and Carlo van Bernem from GKSS for giving me the possibility to work on this project. I would like to thank Ulrich Callies for his constant and fast feedback every time I needed. Our discussions via email or in person in Geesthacht helped me to better understand the problem and its implications in the real life.

I would like to thank again Roger Cooke and Dorota Kurowicka for their good care for the well-being of the students in our group. I would like to thank our friends here in Delft who where close to us when we needed help.

Finally, I would like to thank my wife for her continuous understanding and enormous support. She was always there to help keeping my spirit high and go through all the difficulties. This work is dedicated to her with all my love.

Chapter 1

INTRODUCTION

1.1 Worldwide trends of oil exploitation

Since its first commercial exploitation in Pennsylvania, U.S.A., in 1859 oil has rapidly become one of the most important commodities in the life of modern society. Even though at the beginning of the 20th century it supplied only 4% of the world's energy, few decades later it became the most important energy source. Today oil is the main source of energy and the raw material in the manufacturing of many useful products such as plastics and fertilizers. About 40% of the world's energy and 96% of its transportation energy is supplied by oil. Statistics are showing that until now the world has consumed almost half of the total oil reserves, that is over 875 billion barrels. Another 1,000 billion barrels of proved and probable reserves remain to be recovered[1]. In the following picture the reserves of oil are showed on geographical bases. We can see that there is an acute geographical concentration of the oil reserves as 65% of them are in the Middle East. This makes oil transportation one of the most important issues in the oil business.

[1] Source: http://www.iags.org/futureofoil.html

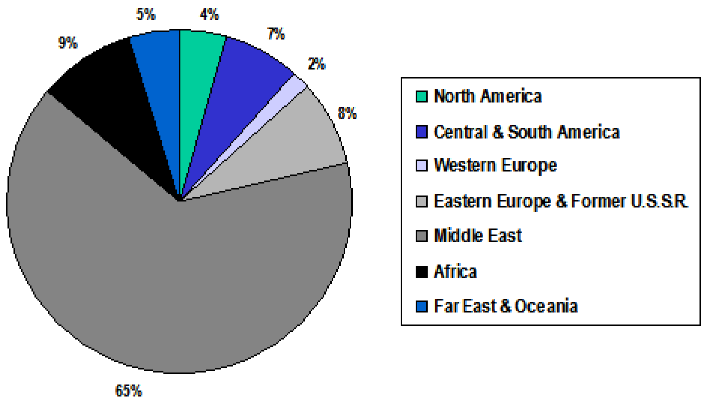

Figure 1.1: Major Crude Oil Reserves, 2003[2]

The same statistics show that in the last decades oil consumption has grown exponentially and will continue to grow at an extremely high rate. It is expected that from now to 2020, world oil consumption rate will rise with about 60%. Transportation will be the fastest growing oil-consuming sector. For example by 2025, the number of cars will increase to more than 1.25 billion from approximately 700 million today, which will probably make the global consumption of gasoline double. The next picture shows the oil consumption from 1978 to 2003 on geographical bases.

[2] Source: http://www.people.hofstra.edu/geotrans/eng/ch5en/appl5en/ch5a1en.html

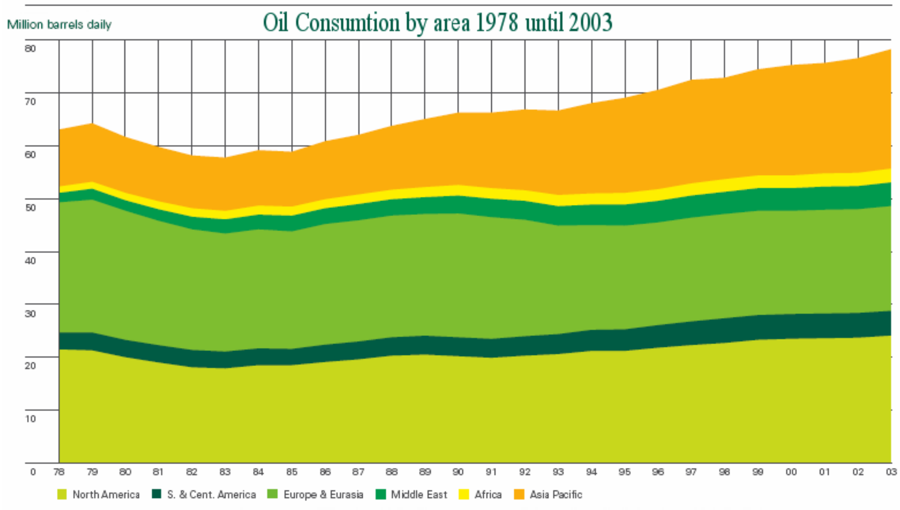

Figure 1.2: Oil Consumption by Area

1.2 Oil Pollution

Lately people are more concerned of how their activity influences the environment and ecosystems. Pollution with oil has one of the most devastating effects on the environment mankind have ever encountered. As we have seen before, because of the geographical concentration of the oil resources, its exploitation requires transportation on vast distances. As oil transportation happens mainly on sea makes the maritime environment be the most affected by oil pollution. The sea is also a very important means of transportation of other goods from which some can be very toxic for the maritime environment. As we can see in the next picture the oil pollution of seas has many sources some of which are at the first thought quite unimaginable.

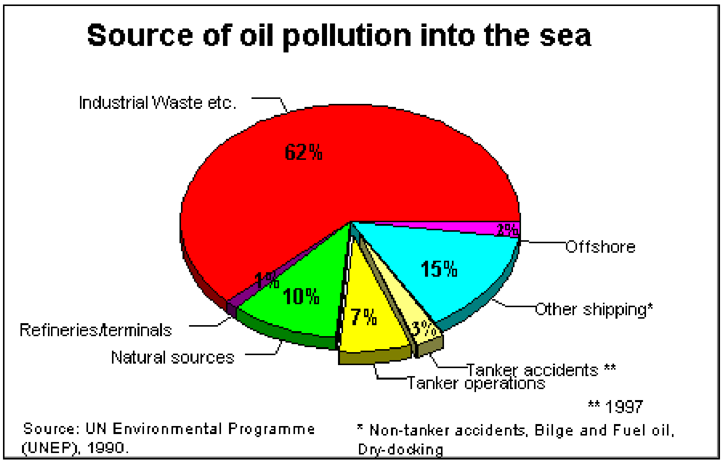

Figure 1.3: Source of oil pollution into the sea[3]

It seams that industrial wastes are the most harmful to the maritime environment since they account for 62% of the oil spilled in the sea. Even though the pollution caused by accidents involving big tankers represents only 3% of the total pollution with oil, the fact that the release of oil is extremely concentrated (in an interval of few days a large amount of oil is released in the water) makes the damages caused by this type of pollution extremely big. The pictures below show two of the largest tankers ever built, so that the reader can get an idea about the quantity of oil that can be shipped nowadays. The tanker in the left is called Jahre Viking and is 458m long, 69m wide and can carry 650,000m^3 (4.1 million barrels) of crude oil. The right one is a typical very large carrier and is 359m long, 60m wide and 30m deep.

[3] Source: "The International Association of Independent Tanker Owners", http://www.intertanko.com/tankerfacts/sizes/pollution.htm

Figure 1.4: The largest tankers ever built

It is not very surprising that this type of pollution, i.e. oil pollution after a tanker accident, gets the most attention from the people. A series of tanker accidents made people realize the need for adequate contingency planning against oil transport accidents. The list of some of the worst oil tanker accidents are presented below:

1. In March 1967, the first super tanker disaster: Torrey Canyon ran aground near Lands End, U.K., spilling 119,000 tons of oil and causing the pollution of Cornwall's coasts.
2. In 1974, tanker Metula spilled 56,000 tons of oil on South American shores.
3. In May 1976 Urquiola ran on rocks near the port of La Coruña, Spain, spilling 107,000 tons of oil.
4. In March 1978, Amoco Cadiz with 220,000 tons of light Arabian crude ran aground off Brittany's coast, polluting the Breton beaches.
5. In March 1989, Exxon Valdez lost 54,000 tons in a remote but vulnerable marine environment in Alaska. This was the worst domestic oil spill in the United States. As a response to this oil spill the US Congress passed the "Oil Pollution Act of 1990" to prevent further spills in the United Stated. It seems that this law was stating: "A company cannot ship oil into the United States until it presents a

plan to prevent spills that may occur. It must also have a detailed containment and cleanup plan in case of an oil spill emergency." [4]

6. In 1996, Sea Empress spilled 70,000 tons in the Irish Sea.
7. In 1999, Erika spilled 20,000 tons 30nm SW from the French port Brest.

1.3 The Wadden Sea, the wind parks, and the problem

The Wadden Sea (Wattenmeer in German, Waddenzee in Dutch and Vadehavet in Danish) lies between a section of the coast of continental Europe and the North Sea. It extends from Den Helder in The Netherlands in the southwest, encloses the river estuaries of Germany at north, to Esbjerg in Denmark[5]. The Wadden Sea is famous for its unique ecosystem. In order to protect this ecosystem, starting in 1978 The Netherlands, Germany and Norway decided to create "The Trilateral Wadden Sea Cooperation" organization. The Wadden Sea area as stated in The Trilateral Cooperation Area is shows in the following map.

[4] Source: "Wikipedia, The free encyclopedia" http://en.wikipedia.org/wiki/Exxon_Valdez_oil_spill

[5] Source: "Wikipedia, The free encyclopedia" http://en.wikipedia.org/wiki/Wadden_Sea

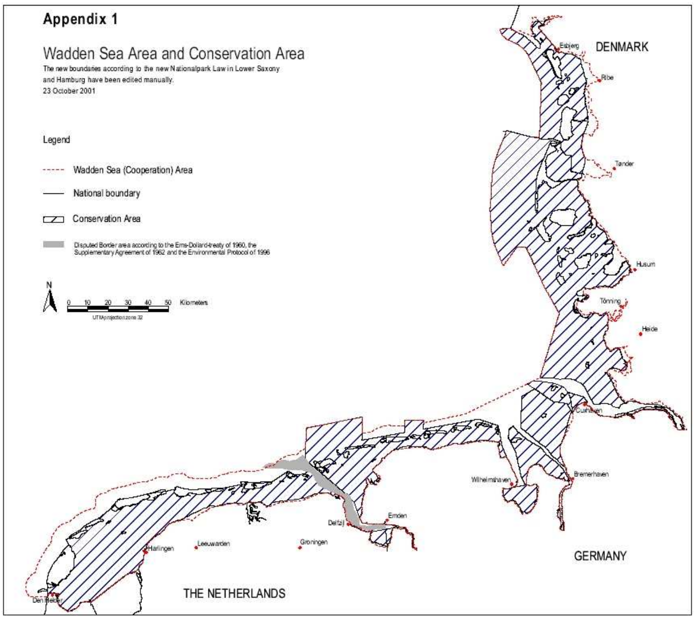

Figure 1.5: The Wadden Sea[6]

Following a trend that appeared all over the world in the last decade, namely to support renewable energy sources[7], the German authorities are planning to build offshore windmill farms in the German Bight. Now, the general concern is that placing the windmill farms in this region will increase the risk of having an accident implicating one of the tankers from the two shipping routes that are crossing the area. The wind farms are planed to be placed in the space between the two shipping routes as is shown in the picture 1.6. The red blocks in this picture are showing wind farms that have already been built or the ones for which the construction has already been approved. The yellow small blocks are the pilot farms already built. The bight was also split into 25 areas of vulnerability showed by the empty blocks.

[6] Source: "The Trilateral Cooperation on the Protection of The Wadden Sea" http://www.waddensea-secretariat.org/index.html

[7] By renewable energy people mean solar energy, wind energy, hydraulic energy, etc. In the last decade people all over the Earth started searching for ecological means to create energy. According to BP statistics wind power generation capacity at the world level is growing at an average rate of 30% per year.

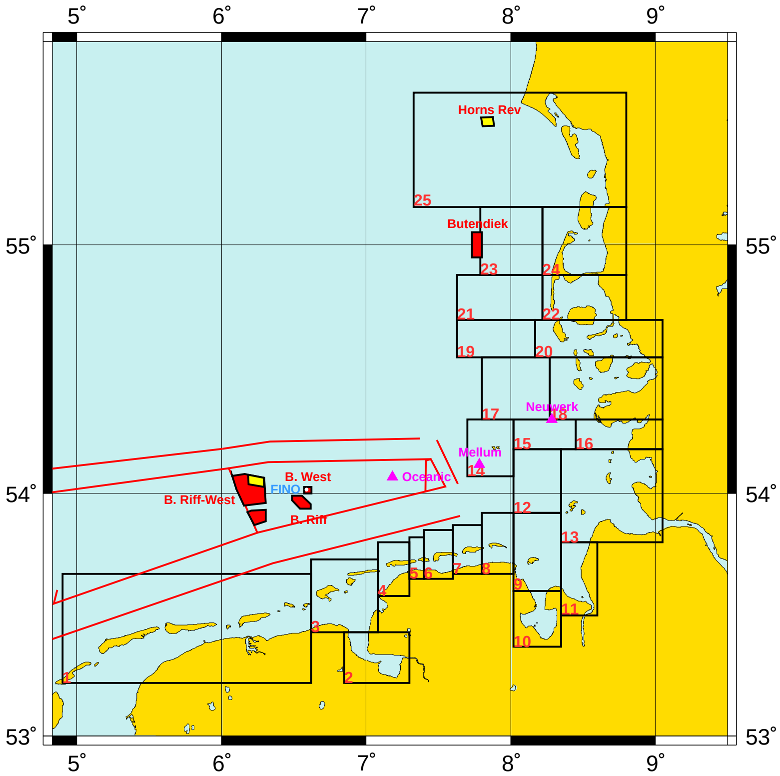

Figure 1.6: German Bight: shipping routes, existing wind farms and sensitivity map[8]

So the present project initiated by GKSS[9] aims to describe the consequences of placing the windmill farms in the given area on the ecosystem of the region.

1.4 Outline of the thesis

For this project the analysis is performed in 2 directions. Each direction tries to answer to one of the questions below:

---

[8] Routes delimitations, the positions and sizes of the planned wind park and the sensitivity map were supplied by GKSS

[9] The GKSS' website: http://www.gkss.de/index_e_js.html

1. How will, the probability of having an accident increase because of the wind plants?
2. How will the consequences of an accident in the region change?

GKSS developed the following diagram in order to direct the analysis inside the project.

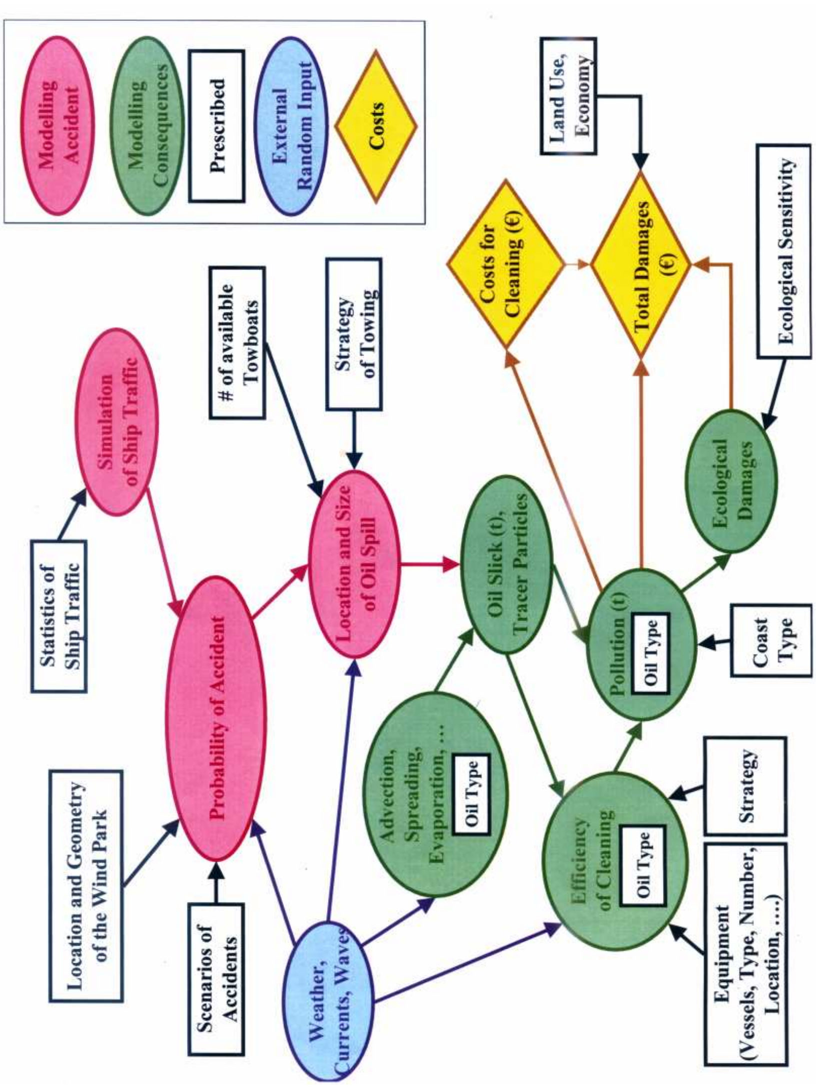

Figure 1.7: Diagram showing the tasks of the project

All the analyses are based on the weather data that is provided by GKSS. The weather data is composed of the wind, wave and current data that were acquired on a period of 42 years on an hourly basis. In the next chapter the weather database will be fully described.

The present thesis is concerned with the red part on the above diagram and tries to compute the probability of having an accident with the windmills park. It was agreed that the accidents we should deal with happen in accord with the following scenario: A ship that travels on one of the shipping routes has a *Loss of Command*[10] at some point on its route and start drifting. After some time if no rescue measures are taken or if the measures are not successful the ship can collide with one of the wind plants. Even though a direct collision without having a Loss of Command is not impossible (it might be caused just by negligence or extremely bad weather), the fact that the windmill plants will be build at some safe distance from the shipping routes makes the probability of such an event very low.

In chapter 2 the weather database used will be described. Also here the complete description of the German "Spill Response Fleet" will be given.

One task of this thesis is to analyze the emergency responses available before the accident, namely the use of towboats to stop the drifting and prevent the accident. Hence the third chapter will present a model that tries to quantify the efficiency of the towboats in the case of a Loss of Command.

The analysis in chapter 3 starts from the assumption that a boat that has a LoC will drift into the wind park if no rescue measures are taken or these measures are unsuccessful. Of course this is not always the case, since depending on the weather conditions the drifting trajectory might drift in other direction than the wind park direction and may have a soft grounding after some time. In chapter 4 we will present a model that aims to quantify the probability to collide with the wind park given that the boat had a LoC.

---

[10] By *Loss of Command* we mean any situation that results in a complete lost of maneuverability of the ship.

The probability to have an accident as a consequence of the construction of the wind park Borkum Riffgrund West, based on the results from the chapters 3 and 4, is given in the chapter 5.

Some comments and conclusions will be presented in chapter six.

In the appendix A, found at the end of this report, the Matlab code used in the analysis will be included with complete explanation.

Appendix B contains the towboats complete database that was used in the simulation.

In Appendix C there will be the lists with "impossible events" for each case analyzed. The definition of "impossible events" can be found in chapter 3.

The graphs that appear in this paper are made using GMT[11] and Matlab[12].

---

[11] The Generic Mapping Tools (GMT) is an open source collection of tools for manipulating geographic and Cartesian data. More information about GMT can be found at: http://gmt.soest.hawaii.edu/

[12] Matlab is a product of The MathWorks: http://www.mathworks.com/index.html

Chapter 2

DESCRIPTION OF THE DATA

2.1 Weather data

The weather database was provided by GKSS and consists of a high-resolution (in time and space) hindcast of wind, waves and currents over a period of 44 years, from 1958 until 2001, for the North Sea. The database was developed during the HIPOCAS, "Hindcast of Dynamic Process of the Ocean and Coastal Areas of Europe", project. The Energy, Environment and Sustainable Development Programme of the European Commission funded the project that run between 01.06.2000 and 31.05.2003. The aim of the project was to produce consistent high-resolution hindcasts of wind, wave and surges over a large period of time, more than 40 years, for all European waters and costal seas. Specific studies were made for the North Sea, North East Atlantic, Mediterranean Sea, Black Sea and Baltic Sea. More about HIPOCAS project can be found on the following websites: http://nokis.baw.de/npublic/research/EU/EVK2-CT-1999-00038.htm, http://www.gkss.de/euprojekte/PSP5/HIPOCAS.html. The following picture shows the project area of the HIPOCAS project.

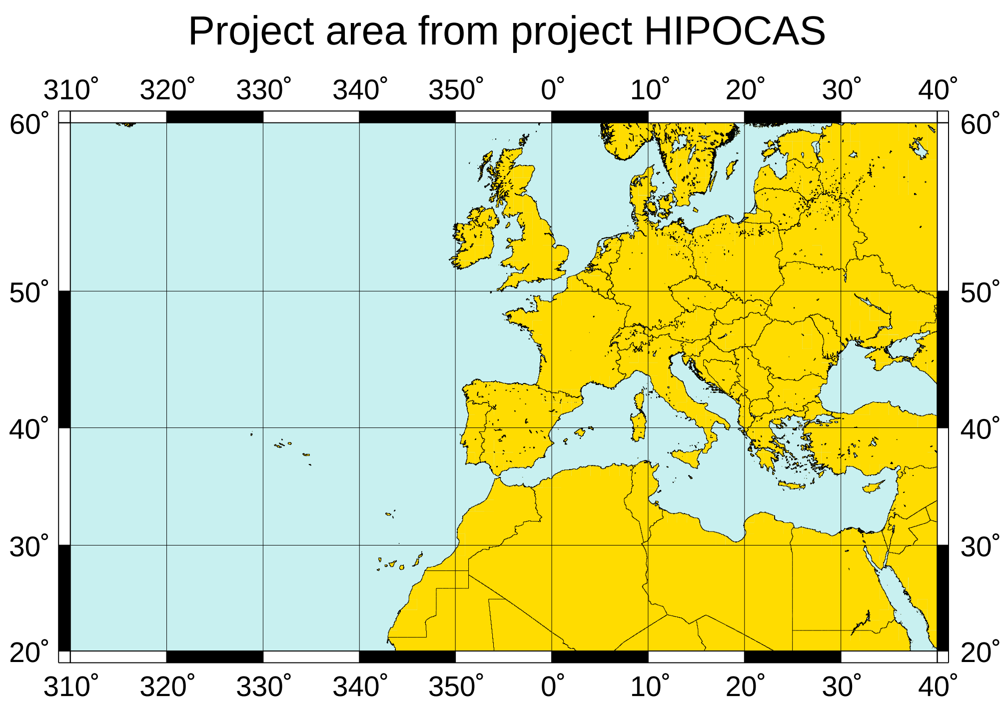

Figure 2.8: Project area from project HIPOCAS[13]

For the North Sea the database has the following content and accuracy: hourly wind data on a mesh size of approximately 50km, tidal data with a spatial resolution ranging from a few hundred meters in the German Bight to a few kilometers in the North Sea, hourly currents data and hourly wave data having also a variable spatial resolution ranging from about 5km south of 56˚N to 50km in the far North. A sample of the mesh used for the Wadden Sea is shown below.

[13] Source: http://w3g.gkss.de/lotse/out/one/meta/HIPOCAS.html

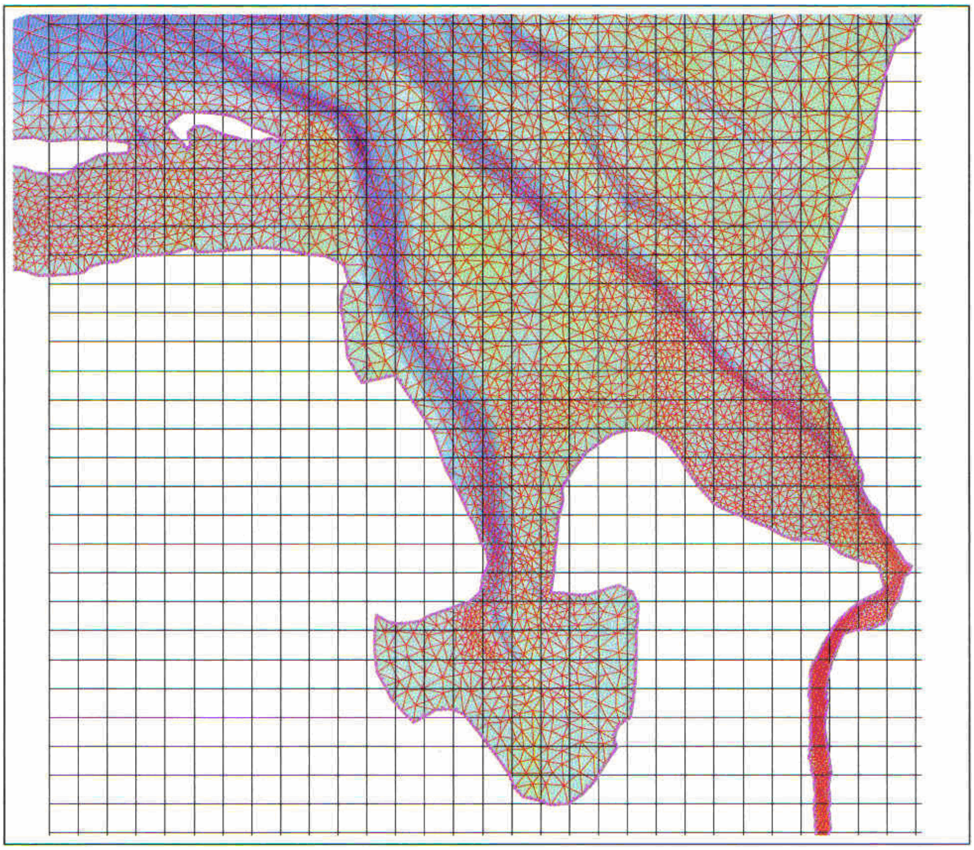

Figure 2.9: Sample mesh used for the Wadden Sea

2.2 Towboats data

Before starting the analysis the database with the German spill response fleet had to be created. The data was collected partially from the Community Information System (CIS), a web based system of the European Union[14]. On this web site information about the capacities of response in case of oil spills for all European Union members is presented. We will list only the towboats here since we are only interested in the towboats fleet of the German authorities. There are two towboats in the German state fleet: Mellum and

[14] Source: CIS website: http://europa.eu.int/comm/environment/civil/marin/cis/cis_index.htm. The information about the third towboat, Oceanic, was taken from "Arbeitsgemeinschaft Küstenschutz" ("Joint Cooperation for Costal Protection") website: http://www.kuestenschutz.com/ .

Neuwerk. Additionally the German authorities have also contracted a third private towboat called Oceanic.

We will list below for each of these towboats their main properties that are extremely important for our analysis. A complete description of each towboat will be presented in Appendix B in the end of this report. In Figure 1.6 from page 8 the expected positions of the three towboats are represented in pink.

Figure 2.10: Mellum

Mellum is a multipurpose boat that travels at maximum speed 16.0kn[15] in optimum weather conditions, and at maximum 11.0kn in bad weather. Its pulling power is 1100kN[16]. The boat anchors in the port Wilhelmshaven and its active state[17] location is 54.11N latitude and 7.78E longitude. Its active state is triggered by wind speeds greater or equal than 8Bft[18]. Otherwise Mellum is operating in the whole German North Sea.

---

[15] 1kn (one knot) equals 1 maritime mile per hour, or 1,852m/h

[16] 1kN (one kilonewton) is a unit of force. It is equal to 1000 newtons

[17] By active state we mean that the boat is sailing and is ready for an emergency towing mission. For each towboat there is a weather threshold set. If the weather conditions become worse than this threshold the towboats will go to their established active state position and stay in expectancy

[18] The Beaufort (Bft) scale is an empirical measure for the intensity of the wind based mainly on sea-state or wave conditions.

Figure 2.11: Neuwerk

Neuwerk is also a multipurpose boat. Its maximum speed in optimum weather is 14kn. It achieved a maximum speed of 8.5kn in heavy weather. Its port is Cuxhaven and its active state location is 54.30N latitude and 8.28E longitude. The weather threshold is also 8Bft.

Figure 2.12: Oceanic

Oceanic's primary task is towing. Its sails at a maximum speed of 17.5kn in optimum weather and it achieved more than 12.5kn in heavy weather. This boat has a pulling power of 1800kN. It has a fix location independent of weather conditions, namely 54.06N latitude and 7.18E longitude.

There is also an agreement, between the Dutch and the German authorities, which says that the Dutch towboat Waker stationed in the port Den Helder will be available when the wind speed is more than 5Bft patrolling along the

West Frisian Islands. This towboat was not included in the analysis since it is too distant from our place of interest.

As we could see above the towboats Neuwerk and Mellum are multipurpose boats and they are operating in the whole North Sea. We will consider the possible positions of the two towboats to be randomly distributed, according to a Gaussian distribution having the mean close to the declared position. For the exact numbers please see the Appendix B from the end of this report. The following picture shows the distribution of 1000 possible positions for each of the towboats.

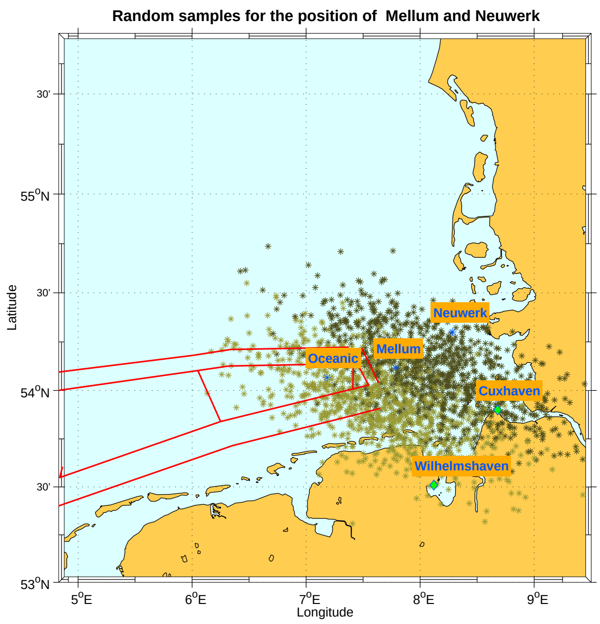

Figure 2.13: Random positions for Mellum and Neuwerk

Chapter 3

THE TOWING MODEL

3.1 Overview

The towing model is in agreement with the scenario presented in the introduction: the accident is supposed to be provoked by a tanker that has a Loss of Command somewhere on its route and starts drifting, and after some time it collides with one of the windmill plants. As we said in the introduction one of the emergency responses is towing. After the Loss of Command the towboat fleet is alarmed. Based on the trajectory estimated with the help of a numerical model, the towboats will try to intercept the drifting boat and then stop it from drifting.

To link the windmill farms positions with the analysis we started from the moment of the accident and then we went back to the moment of the Loss of Command. There were considered two places where a hypothetical accident would have taken place. The first accident location is at 6˚14'25" East longitude and 54˚3'1" North latitude. This place denotes the center of the pilot wind farm already built for the Borkum Riffgrund West wind plant. This accident location is intended to count as a possible accident location for a boat that starts drifting somewhere on the northern route. The second accident location is placed also in the Borkum Riffgrund West but in the southern part of the wind farm. The exact location is: 6˚15'East longitude and 53˚54'30"North latitude. We will use this location to account for accidents caused by drifting ships coming from the southern route.

3.2 Drifting trajectories

A numerical model was developed at GKSS that trace back in time the ship until the moment of Loss of Command. The numerical model is based on a Lagrangian dispersion model that was successfully used for the prediction of the drift and dispersion of oil and other substances in the sea[19]. All the computations are made in accordance with the weather conditions at the moment of the accident and all the way back in time. By placing the accident at a very large number of moments in time, a large database of backward trajectories was constructed.

Below some images with examples of drift trajectories are listed. The trajectory of the drifting boat after it had passed the place of the accident is also presented; that is if the accident would have not taken place. In green we have the forward trajectory and in dark blue the backward trajectory. In pink we have the position of the three towboats in active state, from the German oil spill response fleet. The length in time is 120 hours for both the forward trajectory and backward trajectory. The time and date for each event is written in the upper left corner. We can see that the trajectory of the drifting vessel depends greatly on the weather conditions. The examples below show three totally different situations for each of the two accident locations.

First for the northern location there are: the event on January 2nd, 1995 when the backward trajectory intersects only the northern route, the event on January 6th, 1995 the backward trajectory intersects only the southern trajectory and last example the event on 24th January, 1995 the backward trajectory does not intersects any shipping route in the 120 hours time interval.

[19] A short description of the Lagrangian model can be found here:
http://www.bsh.de/en/Marine%20data/Forecasts/Drift%20forecasts/index.jsp

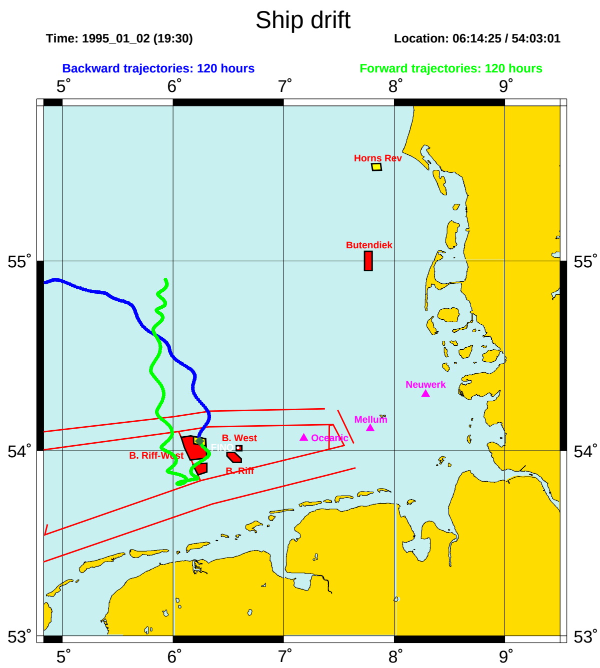

Figure 3.14: Ship drift example 1

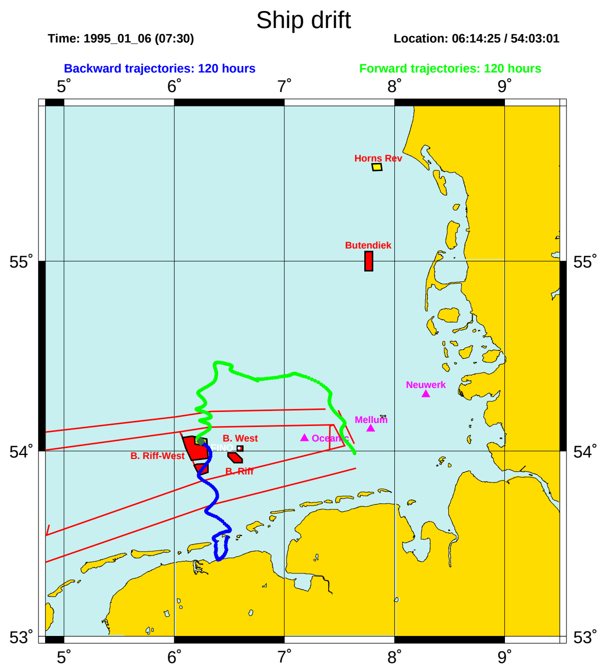

Figure 3.15: Ship drift example 2

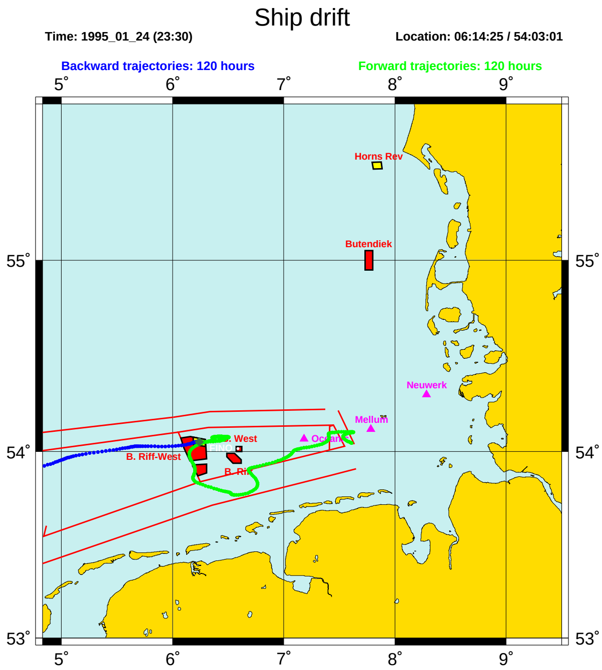

Figure 3.16: Ship drift example 3

If a situation as in the third example appears, we will conclude later in the analysis that the corresponding accident is impossible to occur. We will use the term *impossible event* to denote this type of event.

In the next picture we have all 313 drifting trajectories simulated for the year 1995 for the northern accident.

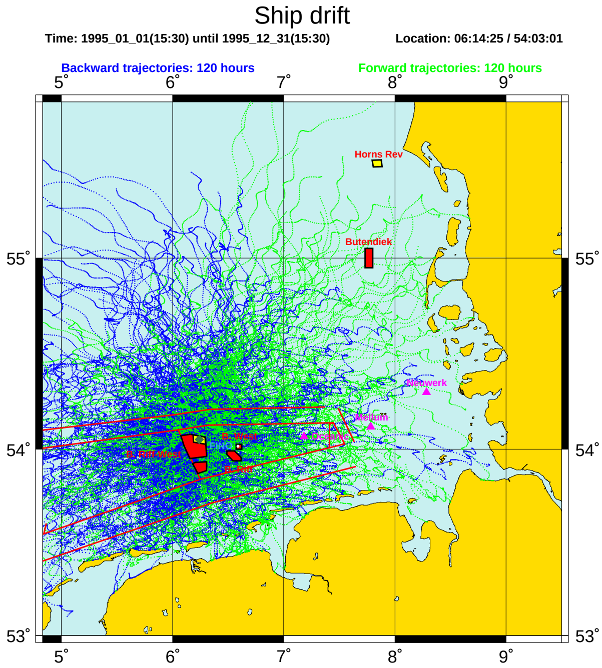

Figure 3.17: All drifting trajectories simulated during the year 1995 for the northern accident location

Now for the southern accident position we will show also three events to exemplify the same three situations that can appear. These events are on January 1st, December 31st and January 25th all in year 1995.

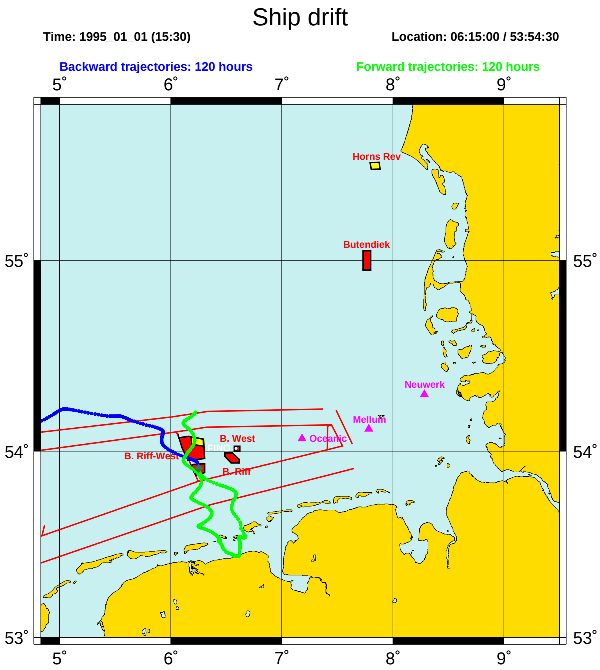

Figure 3.18: Ship drift example 4

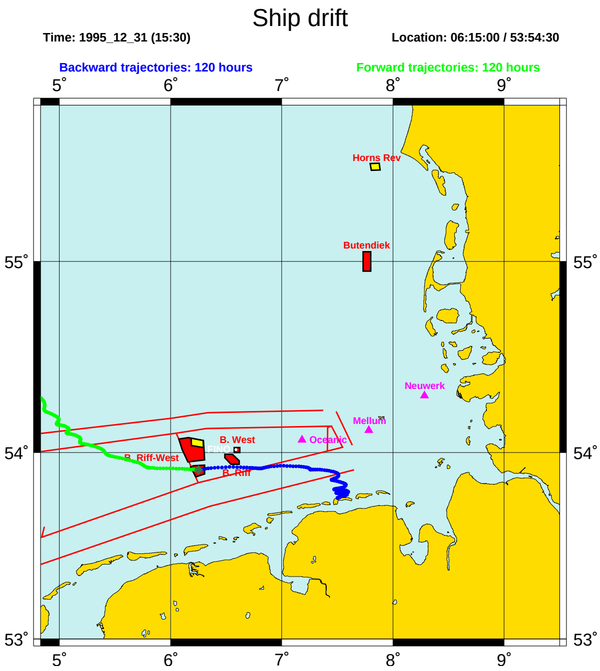

Figure 3.19: Ship drift example 5

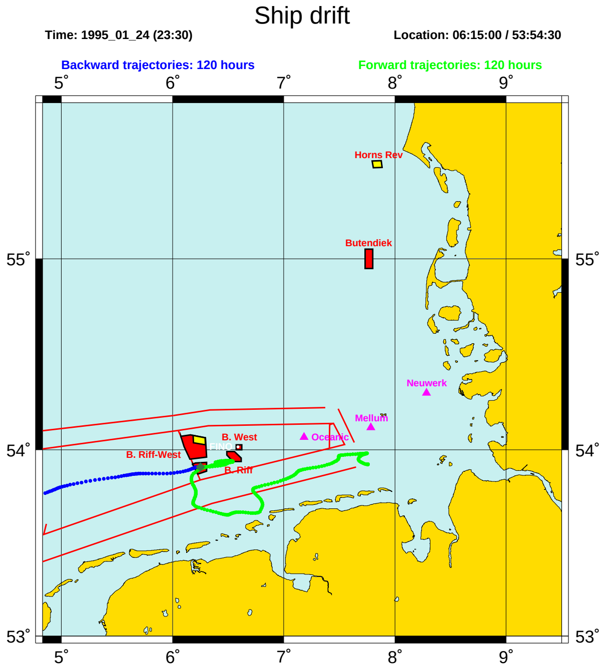

Figure 3.20: Ship drift example 6

The picture with all the trajectories simulated for the year 1995 for the southern accident location will be shown below. As we can see the image with all trajectories for the northern accident location and the one with all the trajectories for the southern accident location are almost identical, the only difference is that all the trajectories are moved to the south. This is to be expected since the same weather conditions are controlling the drifting trajectories.

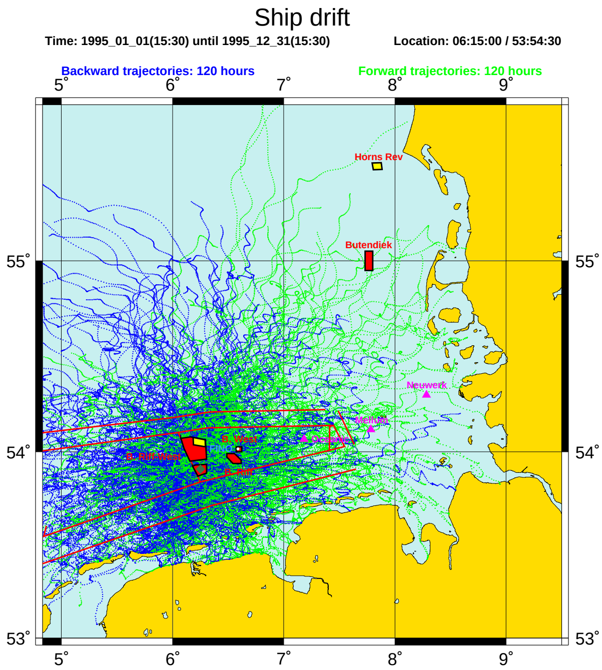

Figure 3.21: All drifting trajectories simulated during the year 1995 for the southern accident location

As we can see in the plot above the backward trajectories start from the accident place and can reach places extremely far from the shipping routes. In their way back, the backward trajectories may or may not intersect one or both shipping routes. They may intersect the routes once or multiple times. As we assume that the ship was traveling on one of the shipping routes before the LoC occurred the place of LoC should be somewhere inside the routes. The question that appeared was when to consider that the LoC happened, at the first intersection of the backward trajectory with the shipping route, or the second, or maybe the last one. The decision was to use as the LoC position the place where the drifting trajectory will cross the route for the first time, since in this way we will be placed on the safe side of the problem, the length of the drifting being the shortest. The results of clipping the trajectories to the first intersection point are shown below. The database used corresponds to the year 1995.

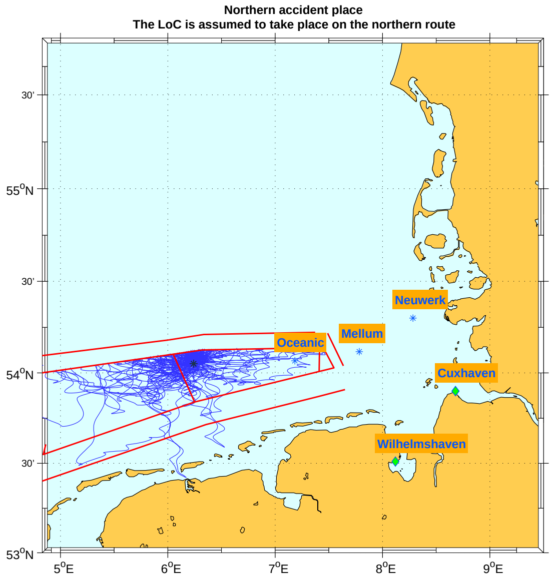

Figure 3.22: Clipped trajectories. Northern accident location. LoC on the northern route

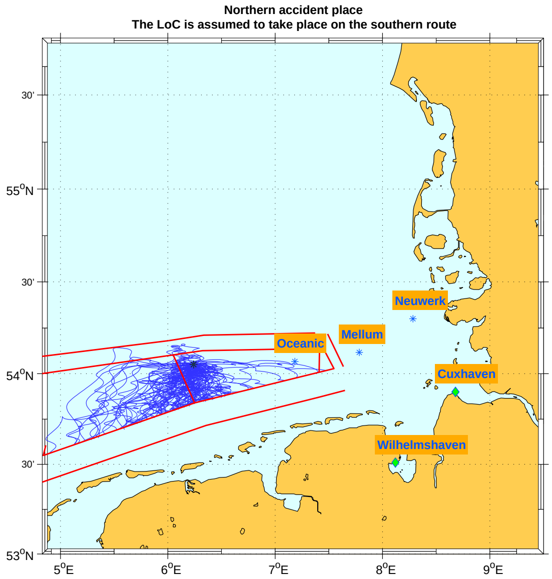

Figure 3.23: Clipped trajectories. Northern accident location. LoC on the southern route

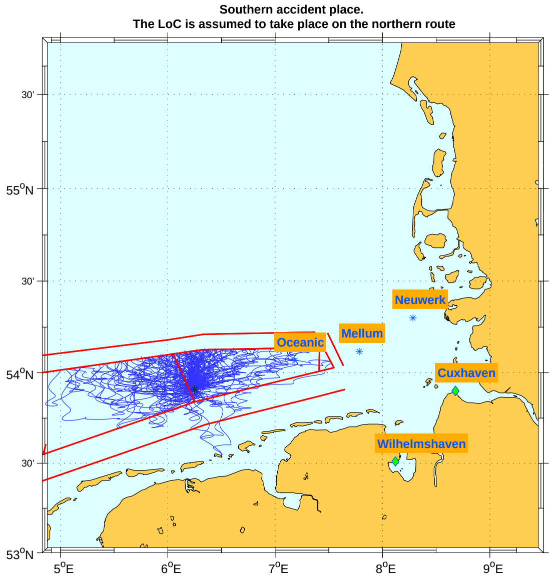

Figure 3.24: Clipped trajectories. Southern accident location. LoC on the northern route

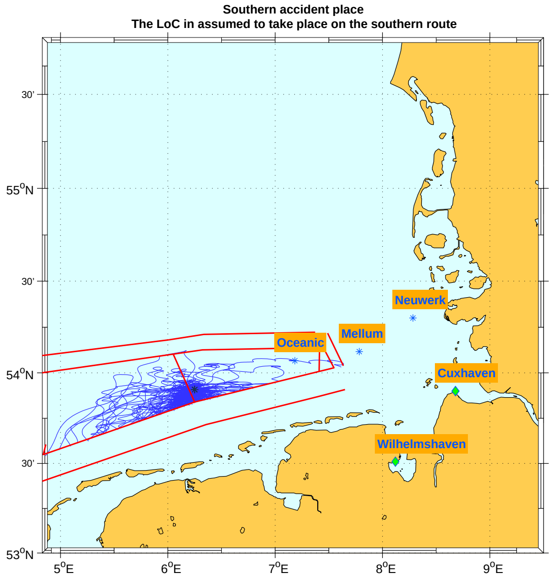

Figure 3.25: Clipped trajectories. Southern accident location. LoC on the southern route

3.3 Detailed description of the model

The towing model is based on the following scenario: when the ship has a Loss of Command and starts drifting it will call for help. The time needed to call for help and get an answer is called *alarm time*. After the towboat has answered to the alarm signal will start preparing the rescue mission. The time needed to prepare the mission is called *react time*. For example in the case of Mellum and Neuwerk if the boats were having some duties far into the North Sea they first have to interrupt their present activities and then be ready to leave. Moreover, sometimes the towing boat needs to go first to its port and unload and maybe change its crew. This situation is however treated separately. When the preparations for the rescue mission are finished an interception based on the trajectory of the drifting boat is attempted. It is not sufficient that the two boats meet but a link should be made between the towboat and the drifting vessel. It seems that linking the two boats is a very difficult action and the time needed is the most uncertain (the experts say that this could last from 20 minutes to 6 hours). The success of this action depends generally on the level of experience of both the towboat crew and the drifting boat crew. Also after the link has been made good experience to stir the towboat in order to bring the drifting boat to a full stop is crucial since the link between the two boats can easily brake. We denoted the time required to make the link *connect time*. The connect time includes also the time needed for reconnection after a broken link. If the connection has succeeded then the time needed to bring the drifting boat to a full stop is required. This is called *full stop time*. So the total time needed to rescue the boat will be computed as:

$$\begin{aligned} total\ time &= alarm\ time + react\ time + \\ &+ bool * go\ to\ port\ time + intercept\ time + \\ &+ connect\ time + full\ stop\ time, \end{aligned}$$

where $bool = 1$ if the towboat needs to go to unload and $bool = 0$ otherwise, *go to port time* is the time needed to go from its present place to port and *intercept time* is the time needed to intercept the boat. If $bool = 1$ then the interception trajectory starts in port.

Now, if the total time is shorter than the time the boat is drifting before the collision, we say that the accident will not take place and the rescue mission is successful.

The alarm, react, connection and full stop times are very uncertain. The analysis will be performed using Monte Carlo methods. We are considering these times as uniformly distributed variables over some given intervals and simulate many times. The following intervals were assumed:

Time	**Range** (hours)	
Alarm time	0.1	0.5
React time	0.1	0.5
Connect time	0.2	6
Full stop time	0.1	1

Table 3.1: The time intervals for the different stages of the towing model

3.4 Results and conclusions

A number of computer programs were developed in Matlab in order to implement the towing model described above. For both locations of the presumed accidents two situations were studied: the case when the boat that collides with the wind park comes from the northern route and the case when the boat comes from the southern route. This was necessary because on the northern route we have very big tankers while on the southern route there are mainly small tankers and other small ships. Of course for the accident that is placed closer to the northern route the results that are more interesting are the ones obtained assuming that the drifting boat comes from the northern route, while for the accident that is placed closer to the southern route the situation is reversed and the most interesting results are the ones that are obtained assuming that the drifting boat comes from the southern route.

The results will present the overall probability of not being able to tow the boat because we are interested in the risk of having a collision, in other words in the bad situations. Also the probability of being unsuccessful to tow the boat will be presented for each event. This is to enable us to weight the consequences of each event. This is extremely helpful if we want to corroborate these results with the results from the consequences modeling side of the project in order to obtain an image of the risk in time. Of course the overall risk can be computed as well using the overall statistics from both analyses.

The mean time needed to rescue the drifting boat from the moment of the LoC and the mean position of the boat at the moment when the boat was brought at full stop are also given for each situation.

Conditioned by the fact that the towing mission has succeeded the probability of being towed by a specific towboat will be also given. This was a question that appeared during the discussions at GKSS, as Oceanic is a towboat rented by the German authorities.

Since the general goal of this project is to quantify the increase in risk as a consequence of building the windmills, a baseline analysis must be performed. In order to present the baseline situation the following analysis was performed: we took the drifting trajectories and instead of stopping them at the location of the presumed collision with the windmills we continued them using the forward trajectory as if the windmill were not there. Doing so we have got drifting trajectories that started somewhere on the routes, where the LoC happened, and ended with a soft grounding near the coast. Of course this will not reflect the entire baseline risk already accepted since we will not include the collision between two ships that are on the route, but we are only interested in accidents that imply that the boat had a LoC. We considered that in the case when the drifting boat is assumed to come from the northern route the soft grounding will happen at the 20m line and for the case when the drifting boat comes from the southern route the soft grounding will happen at the 10m line. These assumptions are in accordance with the size of the ships cruising on the two routes. The resulted trajectories will be presented for each case later in the results.

The results were obtained using the databases for the year 1995. There were 313 events simulated for this year, one at every 28 hours. For each event 1000 trials to save the boat was performed for each towboat.

As we said in the subchapter 3.2 the events for which the backward trajectory never intersects the route under analysis will be called impossible event. Such an event cannot occur since the ship that could provoke the accident should have come from an area where there are no ships. Hence the probability to have such an accident will be zero. In the pictures below these events are also drawn with probability zero to have an unsuccessful towing. For each case the list of impossible events will be presented in appendix C at the end of this report.

3.4.1 Year 1995 - Southern location of accident

3.4.1.1 The ship is drifting from the northern route

There were 141 possible events. Only for the event on 12th January 1995 the probability to tow the boat was different from 1, namely 0.959000. Hence the probability to have an unsuccessful towing was 0.000291. The mean time needed to save the boat was 7.202877 hours with the variance of 3.617317. The mean position was 54.040948N latitude and 6.192754E longitude. The variance of the location was 0.002699 degrees of latitude and 0.228686 degrees of longitude. The probability to be saved by a specific towboat, given that the towing has succeeded is shown in the next picture

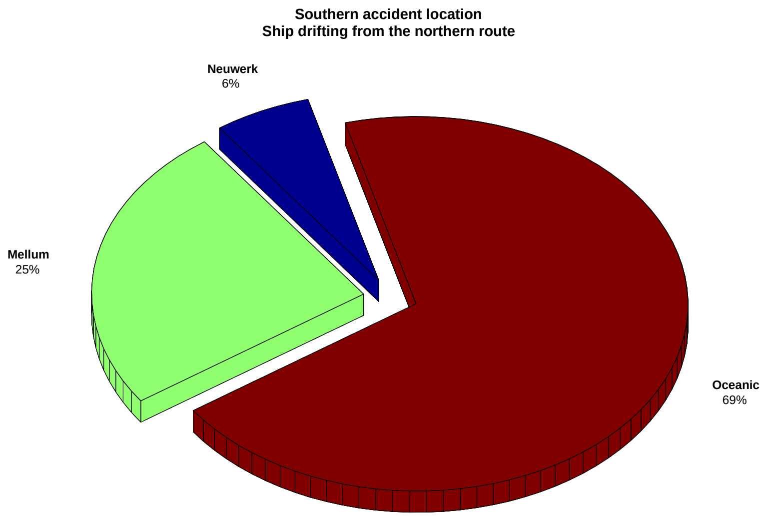

Figure 3.26: The probability to be towed by a specific towboat

3.4.1.2 The ship is drifting from the southern route

For this case there were 160 possible events. The probability to be unsuccessful was 0.125281. The mean time needed to save the boat was 7.418913 hours with a variance of 3.229090 hours. The position at the end of the mission has the following statistics:

- Mean location: [54.855764N, 6.130873E]

- Variance: [0.002588, 0.112660] degrees

The probability of being unsuccessful per each event and the probability to be towed by a specific towboat are shown in the next two pictures:

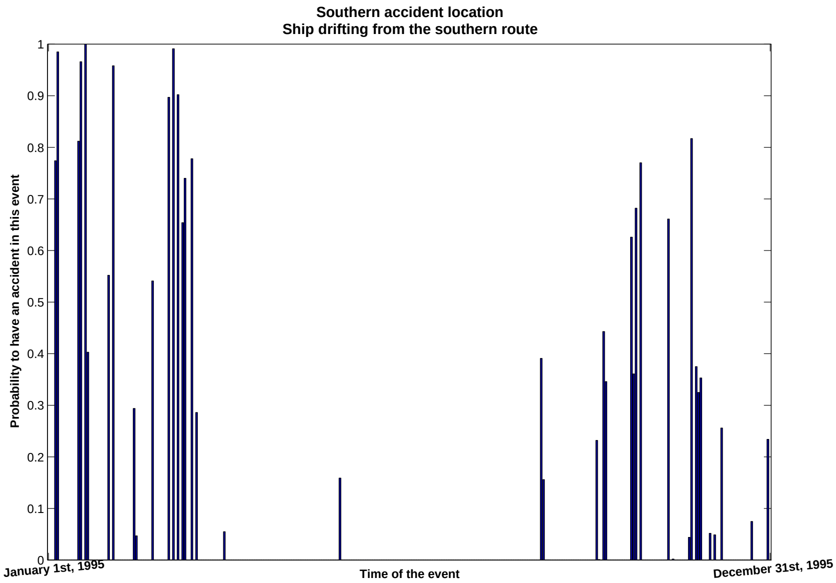

Figure 3.27: The probability of being unsuccessful

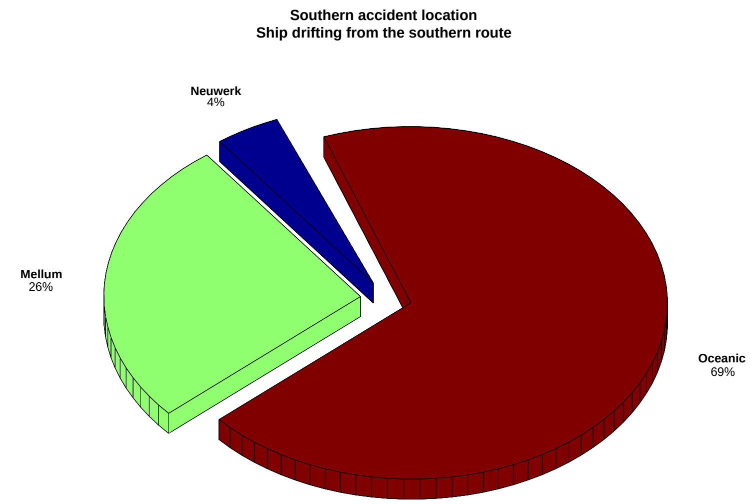

Figure 3.28: The probability to be towed by a specific towboat

3.4.2 Year 1995 - Northern location of accident

3.4.2.1 The ship is drifting from the southern route

Only for one of the 104 possible events found in this case, the probability to be successful differs from 1. That is the event on 20th January 1995 for which the probability to be successful was 0.984000. Hence the overall probability to have an unsuccessful mission for these settings is 0.000154. Then mean time to save the boat was 6.647740 hours with a variance of 3.620147 hours. The location of the boat when the full stop was obtained has the following statistics:

- Mean location: [53.866224N, 6.083962E]

- Variance: [0.005067, 0.183991] degrees

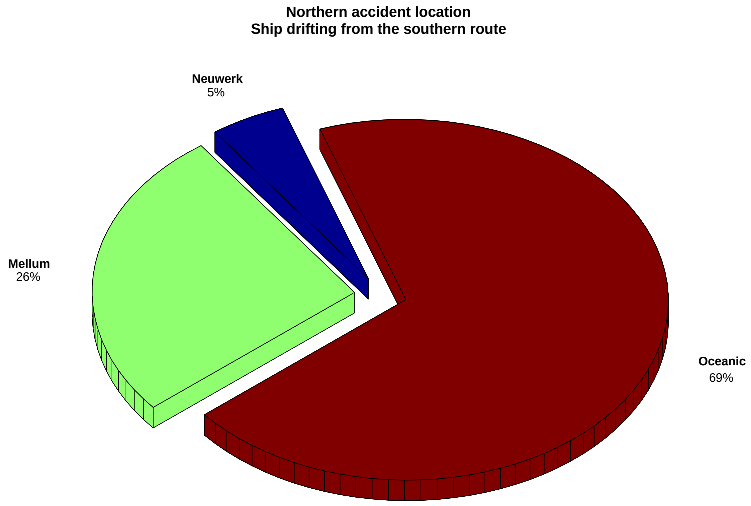

Figure 3.29: The probability to be towed by a specific towboat

3.4.2.2 The ship is drifting from the northern route

There were 190 possible events. The overall probability to be unsuccessful is 0.133421. The mean time to rescue the boat is 7.112871 hours with a variance of 3.422845 hours. The location at the end of the mission has the following statistics:

- Mean location: [53.062441N, 6.179870E]

- Variance: [0.002098, 0.179898] degrees

The probability to be unsuccessful per event and the probability to be towed by a specific towboat are given below.

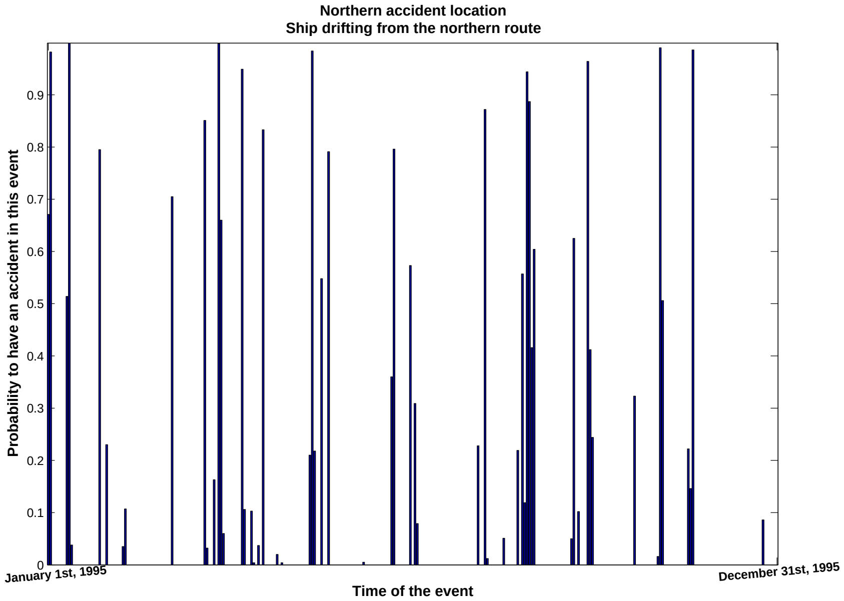

Figure 3.30: The probability of being unsuccessful

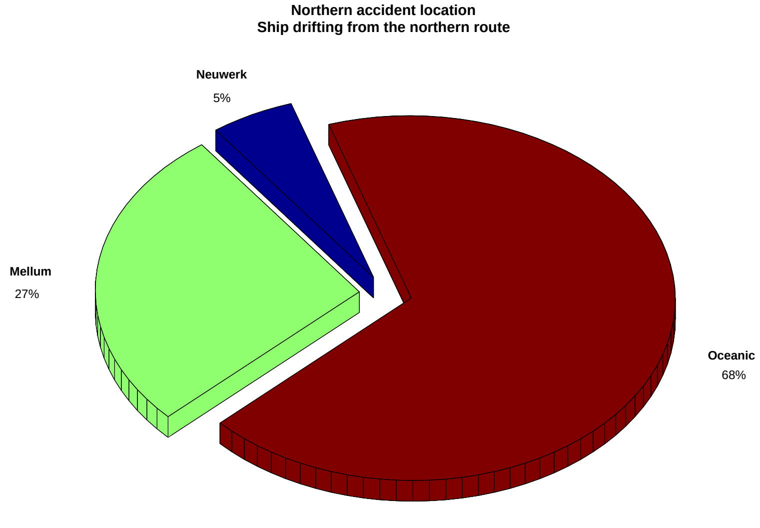

Figure 3.31: The probability to be towed by a specific towboat

3.4.3 Baseline

First of all we will list below the pictures with drifting trajectories obtained for each of the four cases.

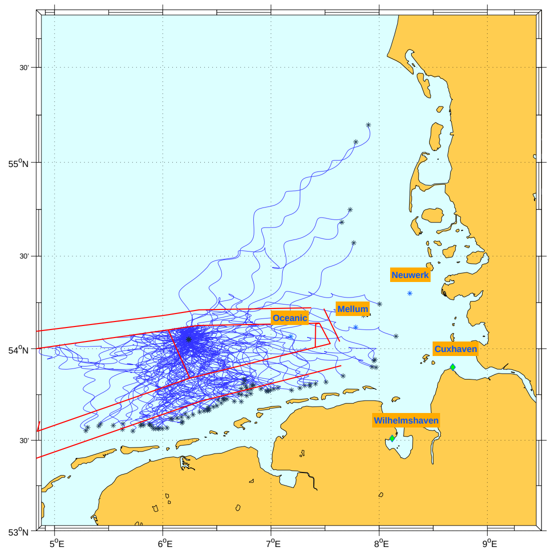

Figure 3.32: Southern accident location. Northern route

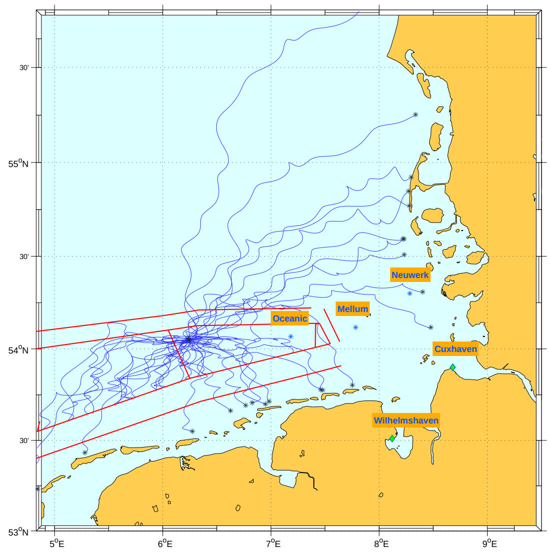

Figure 3.33: Southern accident location. Southern route

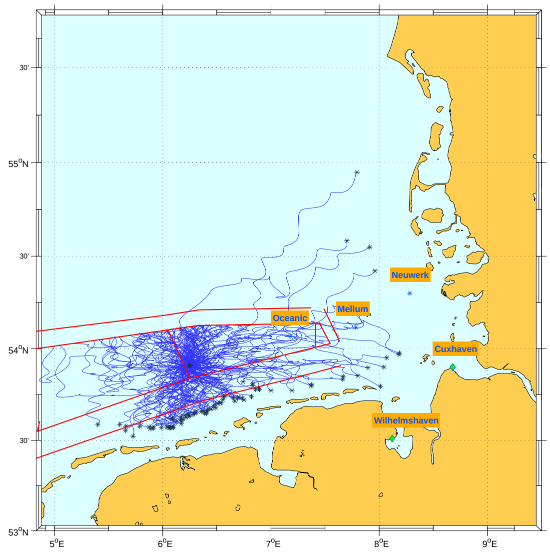

Figure 3.34: Northern accident location. Northern route

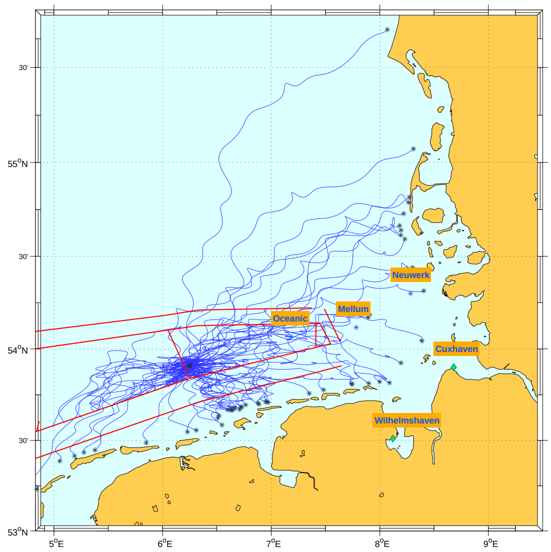

Figure 3.35: Northern accident location. Southern route

In all these cases we obtained that the probability to tow the drifting boat is equal to 1. The other results for the baseline analysis were compiled in the next table:

	Possible events	Mean time (hours)	Time variance (hours)	Mean location (degrees)	Location variance (degrees)
Northern location, northern route	91	7.306653	3.075624	54.054993 6.138196	0.001918 0.108193
Northern location, southern route	21	8.216566	4.414919	53.838330 5.880794	0.008597 0.295760
Southern location, northern route	88	7.190630	3.442353	54.044462 6.188371	0.002866 0.182374
Southern location, southern route	50	7.678290	3.523938	53.859569 6.114792	0.004262 0.178646

Table 3.2: Baseline statistics

The probability to be towed by a specific towboat has the same pattern as in the cases above, with Oceanic on the first place saving 70% of the boats followed by Mellum saving approximately 25% of the drifting boats and finally Neuwerk being the first one to save the boat in 5% of cases. The next pictures show the exact numbers.

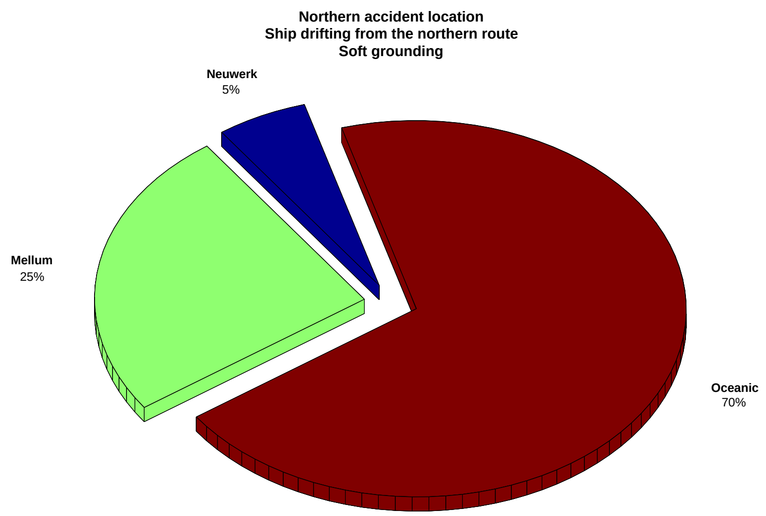

Figure 3.36: The probability to be towed by a specific towboat. Baseline: North-North

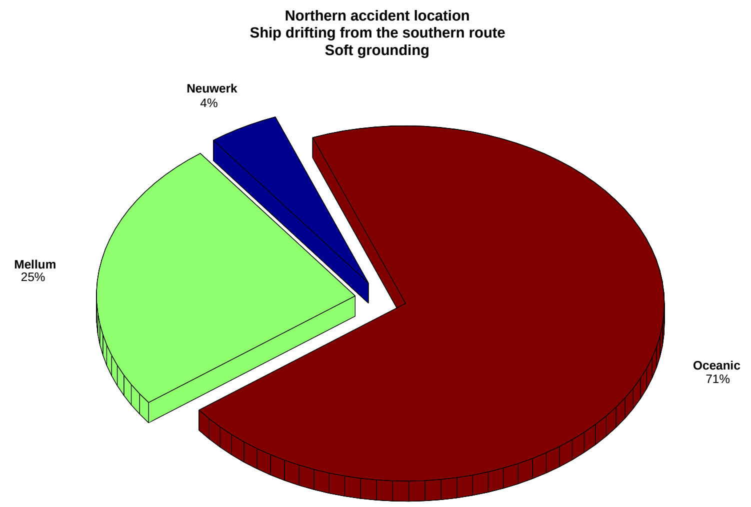

Figure 3.37: The probability to be towed by a specific towboat. Baseline: North-South

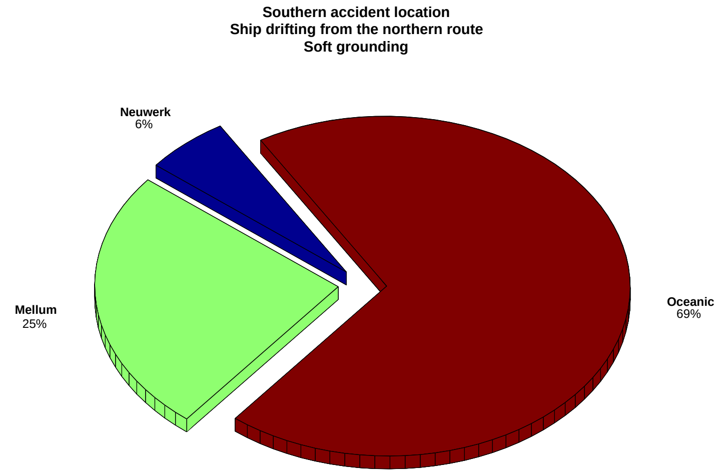

Figure 3.38: The probability to be towed by a specific towboat. Baseline: South-North

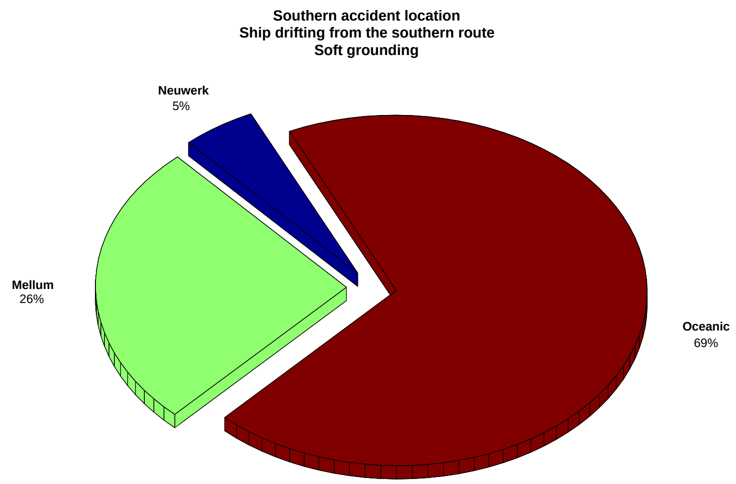

Figure 3.39: The probability to be towed by a specific towboat. Baseline: South-South

3.4.4 Conclusions

We listed the results of the analysis for the year 1995. Also a baseline model was presented together with the results for the year 1995.

The first conclusion that we can draw is that in the case of a Lost of Command having an operative towboat fleet will greatly reduce the risk of having an accident. We can see, in the results above, that is possible to reduce the risk by 91.9% in the worst case to 99.9999% in the best case. Of course a better strategy for placing the towboats in the area of interest is always welcome. We see clearly that the towboat Oceanic will always be more successful in trying to tow the drifting boat. Its main advantage is its position very close to the shipping routes. Also the fact that this position is fixed when the weather becomes worse than a given threshold increases the chances to be rescued by Oceanic. So the probability to be towed by Oceanic is around

70%, the probability to be towed by Mellum is around 25% and the probability to be towed by Neuwerk is around 5%.

An interesting remark is that in the case of the southern accident provoked by a drifting ship coming from the southern route there is a season dependent increase of the risk. Hence almost all "bad" events are clustered in the interval from late autumn until beginning of March. This situation is not visible in the other situation when the accident location is to the north and the drifting ship comes from the northern route. In the next picture the wind vector at every hour in the year 1995 is presented. We can see that in January, February and the first half on March the wind is blowing mainly in northerly direction which makes the drifting trajectories shorter. Therefore the towing mission is more complicated. The same pattern reappears in the autumn. During summer the situation is revered, the wind is blowing mainly southerly or if not for example in June and July the wind speed is very small. The coast, the shipping routes, and the southern accident location are also presented in this picture. The wind measurements are in only one location.

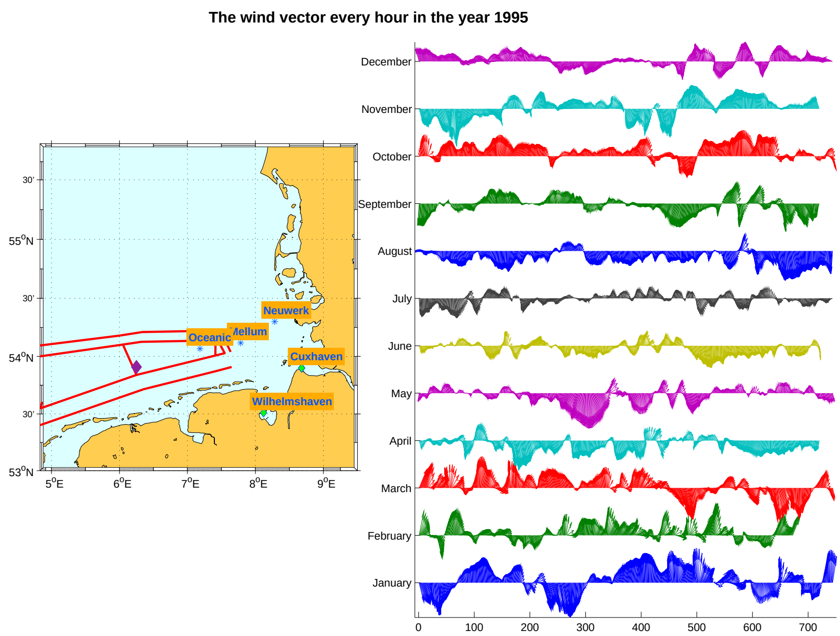

Figure 3.40 The wind vector hourly, during 1995

Chapter 4

PROBABILITY TO DRIFT INTO THE WIND PARK GIVEN THAT THE BOAT HAD A LOC

4.1 Overview

In the previous chapter the analysis started from the moment of the accident and the probability to have a successful towing was computed. In this way we implicitly made the assumption that if a boat has a LoC then it will definitely drift into the wind park. This chapter will present a model for computing the probability to drift into the wind park given that the boat had a LoC.

4.2 The model

We started with splitting the shipping route of interest into a sufficient large number of zones. The center of each zone will be considered a possible place for the LoC. From each of these points at various moments in time the drifting trajectory was computed using the same Lagrangian model presented in chapter three. On each of the two routes, northern and southern, were taken 100 uniformly distributed points. In the next pictures are shown all 200 points, on the northern route with green and on the southern route with red.

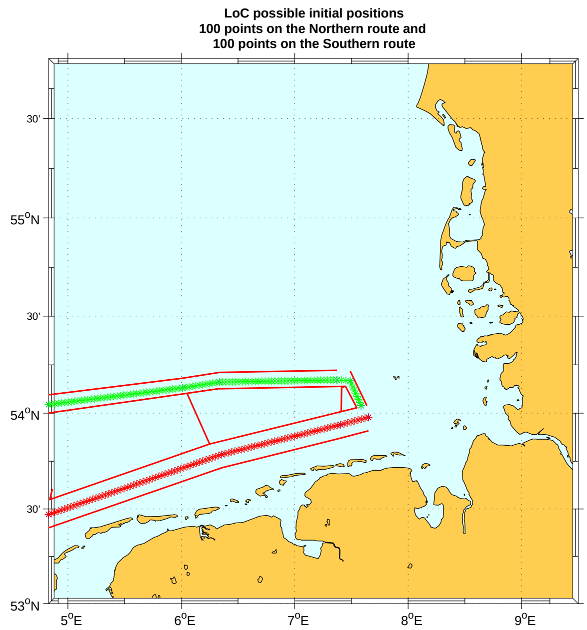

Figure 4.41: LoC possible locations

From each of these points a drifting trajectory was computed every 28 hours. The length of each trajectory is 120 hours. Having a trajectory the next step is to see if the drifting ship will collide with the wind park or not. If the drifting trajectory that starts from $location_i, where\ i \in \{1,2,...,100\}$ at time t intersects the boundary of the wind park of interest then we set $col(i,t)=1$, otherwise $col(i,t)=0$. Of course if the drifting boat goes first to very shallow waters then the analysis of this trajectory stops and a soft grounding is registered.

We define the probability to drift into the wind park given that the boat had a LoC and given the location of the LoC as follows:

$$P(Drift\ to\ wind\ \ park \mid LoC\,\&\,location_i\ LoC) = \underset{time}{Mean}(col(i,time))$$

We define the probability to drift into the wind park given that the boat had a LoC and given the time of the LoC as follows:

$$P(Drift\ to\ wind\ \ park \mid LoC\ \&\ time\ of\ LoC) = \frac{1}{100}\sum_{i} col(i, time)$$

We define the probability to drift into the wind park given that the boat had a LoC as follows:

$$P(Drift\ to\ wind\ \ park \mid LoC) = \underset{time,i}{Mean}(col(i, time)) = \frac{\#\, of\ registered\ collisions}{\#\, of\ events\ simulated}$$

4.3 Results and conclusions

In this section are listed the results obtained from the Matlab implementation of the model.

We performed the analysis for the Borkum Riffgrund West wind park. This wind park is the largest wind park planed to be build and is situated in the west side of the German territorial waters as can be seen in the Figure 1.6. As we can see in the Figure 1.6 this wind park is divided into two parts. In chapter three there were two accidents locations analyzed, one located in the northern part of the wind park, more exactly in the pilot wind park which was built here, and one located in the southern part. Hence the present analysis will be also divided in two cases based on the route where the LoC is situated.

We will present first the results of the analysis obtained using the database for the year 1995 and then the ones obtained using a larger time interval, namely 1990 to 1999.

4.3.1 Year 1995

4.3.1.1 LoC is on the northern route

For this case, when the LoC is assumed to happen on the northern route, we checked for collisions with the northern section of the Borkum Riffgrund West wind park. We will show first the results of the analysis when the whole section is considered and then the results when only the pilot part is considered.

4.3.1.1a Collision with the whole northern section

The probability to drift into the wind park is 0.10105431. The next picture will show the probability to drift into the wind park on geographical bases:

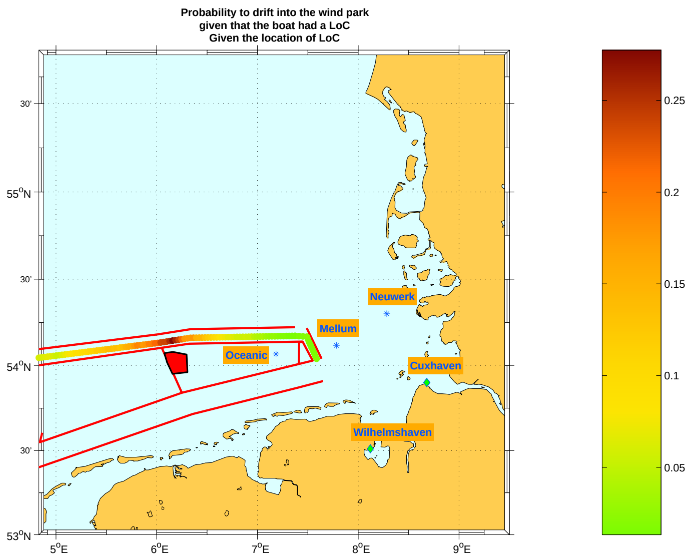

Figure 4.42: Probability to drift into the wind park given LoC and the location of the LoC

We can observe a concentration of the dangerous positions exactly at the north from the wind park. The probability to drift into the wind park as a function of location goes to almost 0.30 at its peak. The same probabilities will be shown in the next picture as bars.

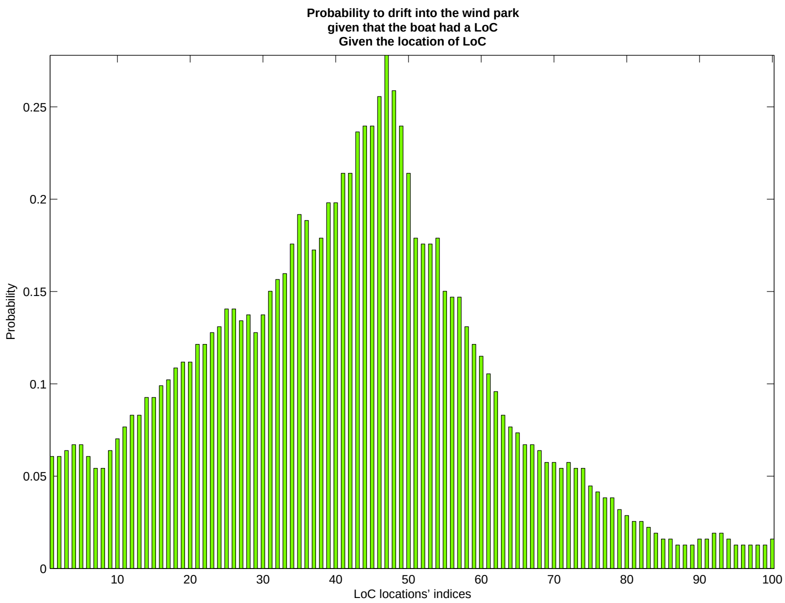

Figure 4.43: Probability to drift into the wind park given LoC and the location of the LoC

We can see in the picture above that in the western side of the wind park the risk of a LoC which can lead to a collision is higher than in the eastern side.

The probability to drift into the wind park given LoC and the moment of the LoC is depicted in the next picture. In this image 1 denotes the event on 1st of January 1995 and 313 denotes the event on 31st December 1995.

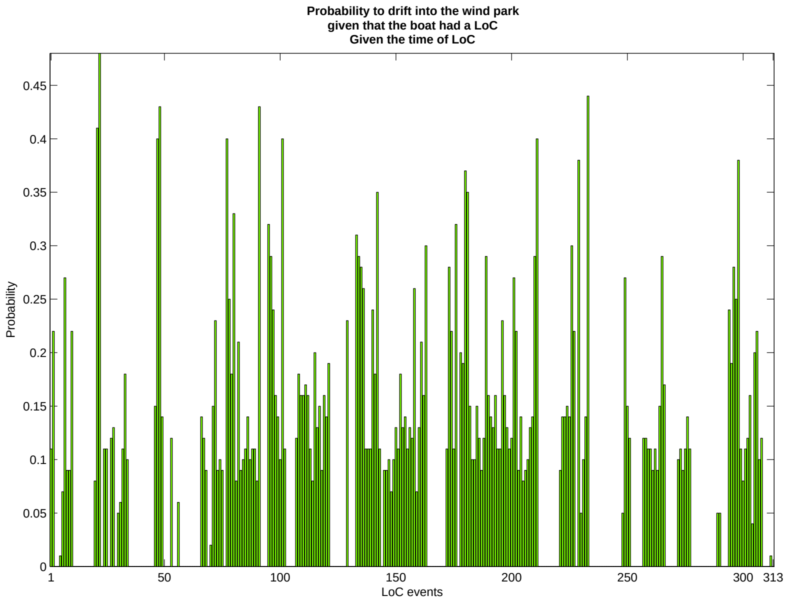

Figure 4.44: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.1.1b Collision with the pilot section from the northern part of the wind park

The probability to drift into the pilot section when the LoC happens somewhere on the northern route is 0.05239617. It is twice smaller then in the case when the whole northern section of the wind park was considered. The probability to drift into the wind park given LoC and location of the LoC is shown in the figures below. The same patterns as above are observed in these pictures too, only that the probabilities are lower.

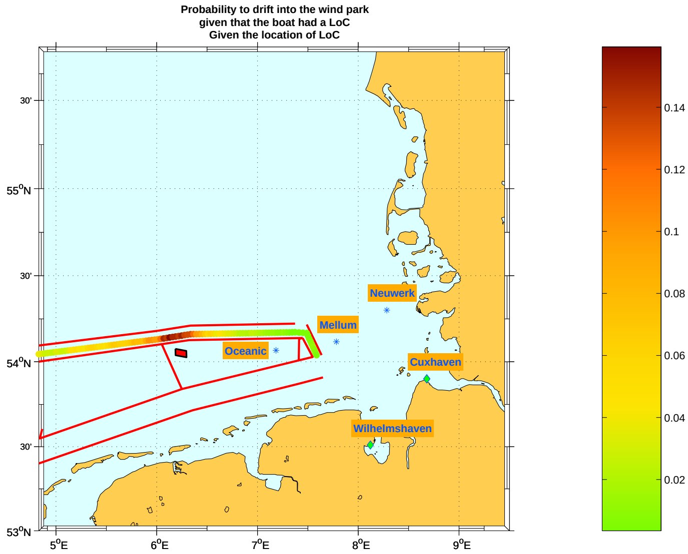

Figure 4.45: Probability to drift into the wind park given LoC and the location of the LoC

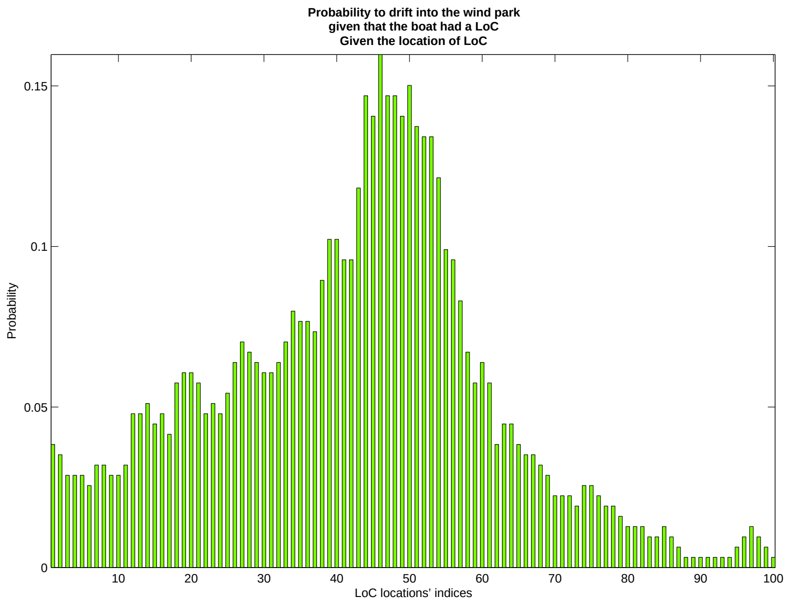

Figure 4.46: Probability to drift into the wind park given LoC and the location of the LoC

The probability to drift into the pilot wind park given a LoC somewhere on the northern route as a function of time is shown below.

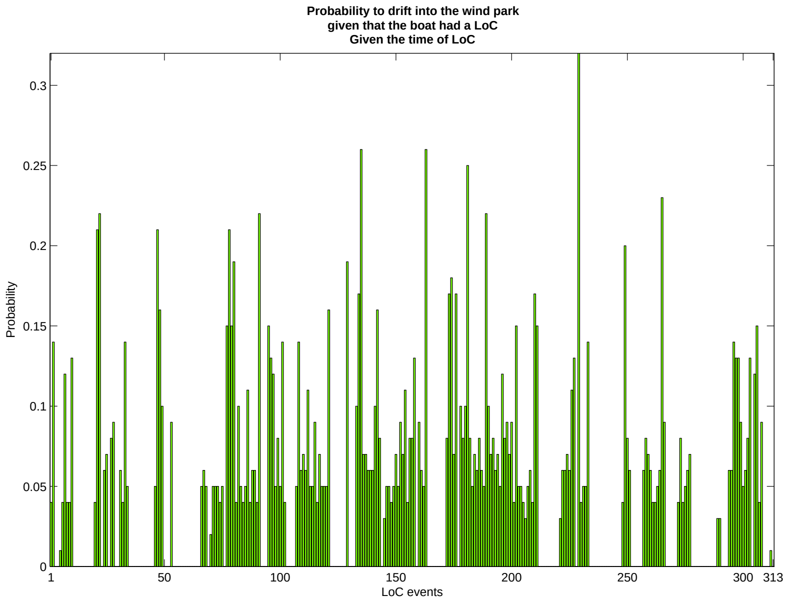

Figure 4.47: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.1.2 LoC is on the southern route

For this case when the LoC is considered on the southern route we searched for collisions with the southern part of the Borkum Riffgrund West wind park.

In this case we found that the overall probability to drift into the wind park given that the ship had a LoC somewhere on the southern route is 0.04348243.

The probability to drift into the wind park given a LoC as a function of geographical location is shown in the next two pictures.

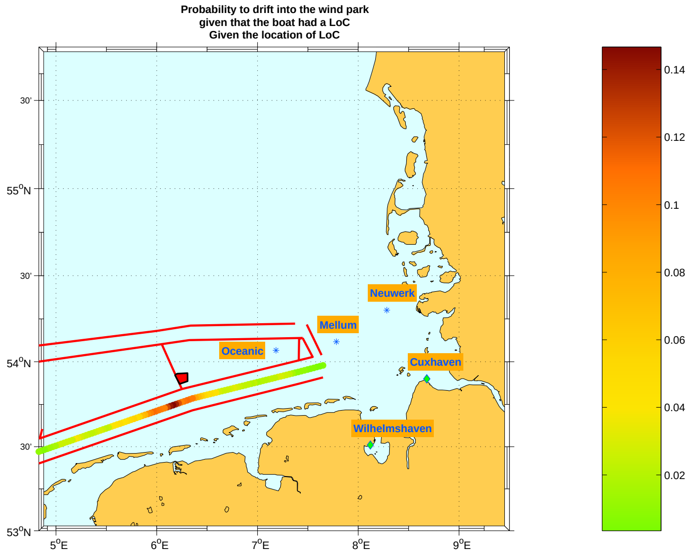

Figure 4.48: Probability to drift into the wind park given LoC and the location of the LoC

The same concentration of dangerous locations as in the northern route case is visible but this time not exactly southern from the wind park but somewhat to the west. However, this time the probabilities are more symmetrically arranged.

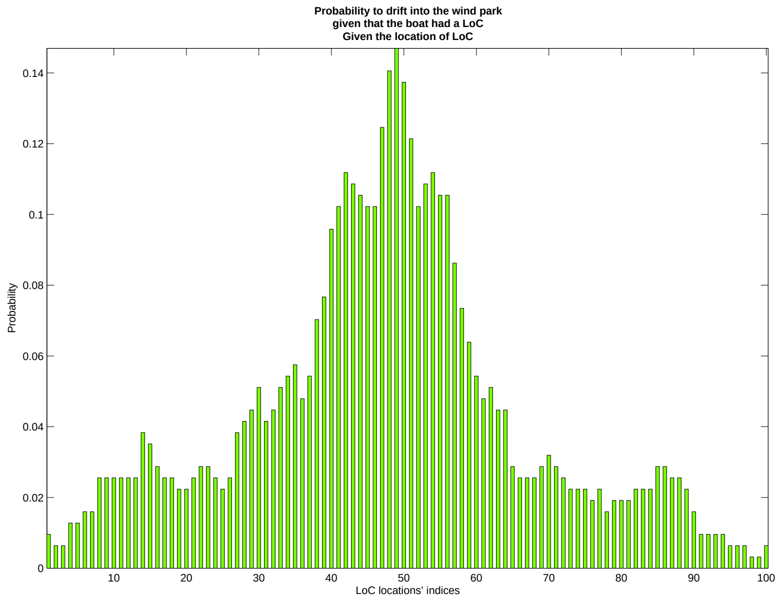

Figure 4.49: Probability to drift into the wind park given LoC and the location of the LoC

The probability to drift into the wind park given the LoC and the moment of LoC is shown in the next picture.

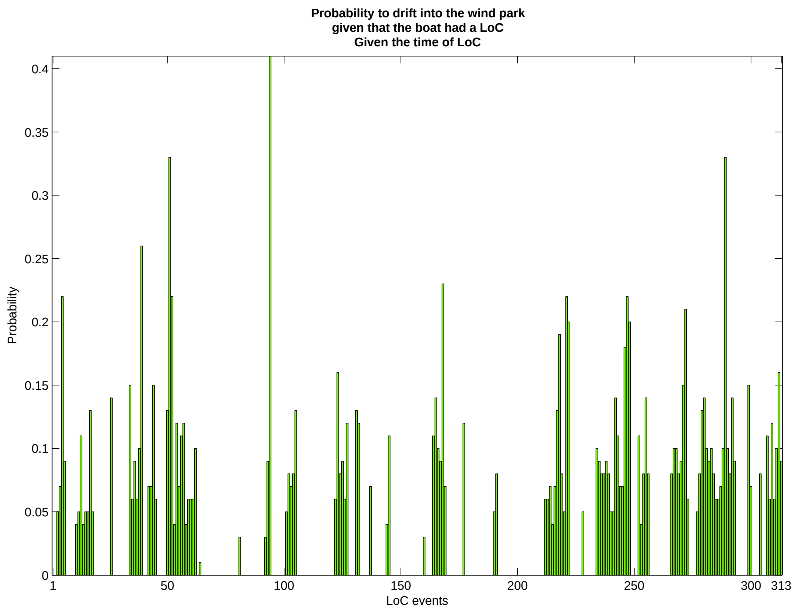

Figure 4.50: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.2 Years 1990 to 1999

Very shortly we will list below the results of the analysis for the period 1990-1999. We will see that the patterns identified above will still govern the results. Of course, having more data, now the curves are smoother. The overall probabilities seem to be just a bit smaller. Since the number of events is now 3130 the plots showing the probabilities per event are less helpful.

The next table shows the overall probability to drift into the wind park for each of the three situations.

Probability to drift into the wind park given that the boat had a LoC Database 1990-1999		
Northern route / Whole wind park	**Northern route / pilot wind park**	**Southern route**
0.08596486	**0.04476038**	**0.05943770**

Table 4.3: Probability to drift into the wind park given LoC

4.3.2.2 LoC is on the southern route

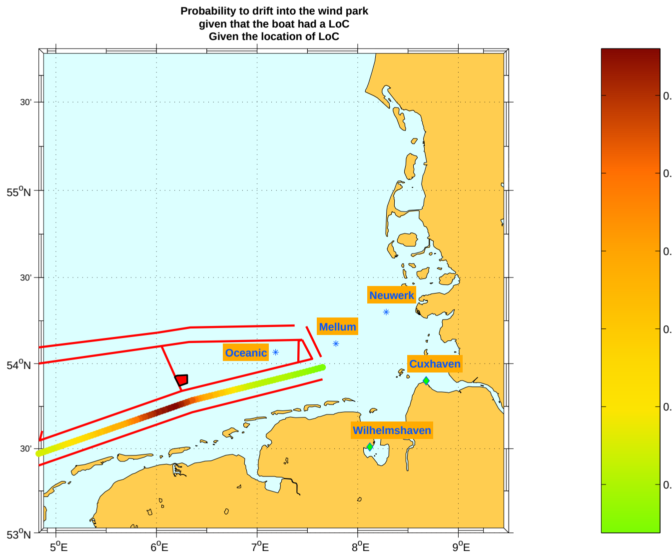

Figure 4.51: Probability to drift into the wind park given LoC and the location of the LoC

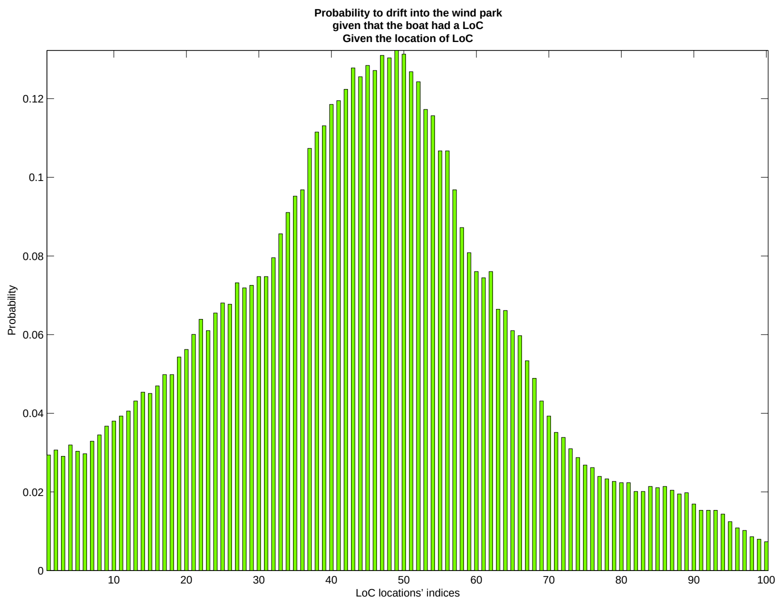

Figure 4.52: Probability to drift into the wind park given LoC and the location of the LoC

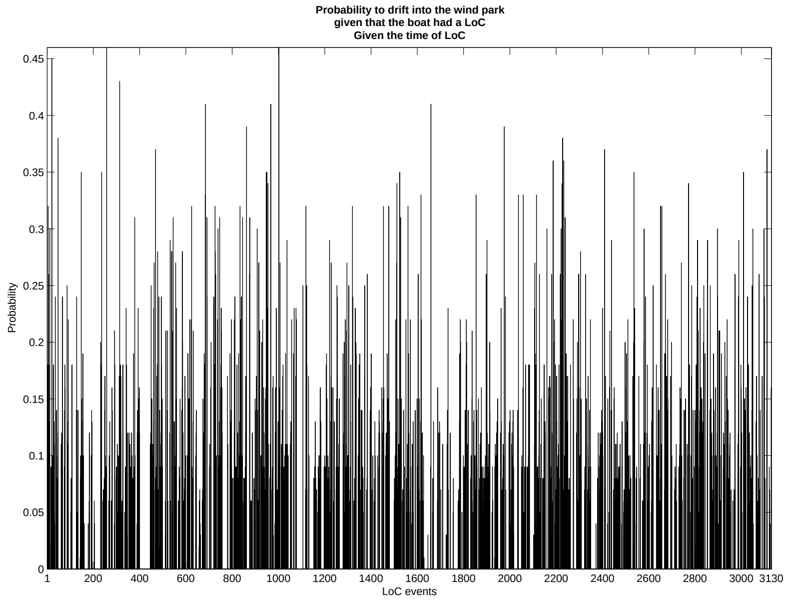

Figure 4.53: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.2.2 LoC is on the northern route

4.3.2.2a Collision with the pilot section from the northern part of the wind park

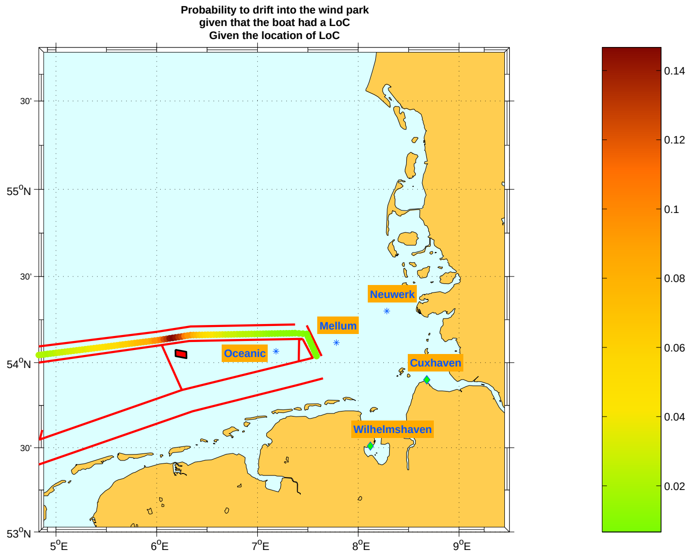

Figure 4.54: Probability to drift into the wind park given LoC and the location of the LoC

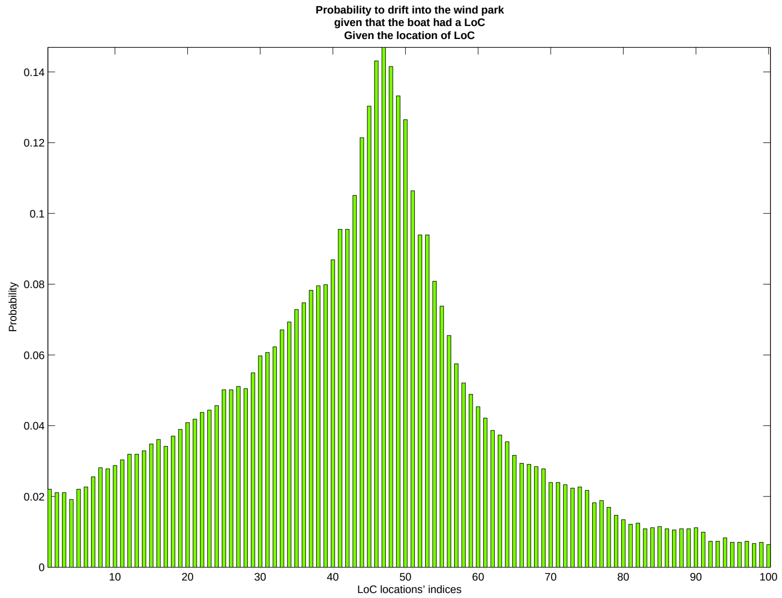

Figure 4.55: Probability to drift into the wind park given LoC and the location of the LoC

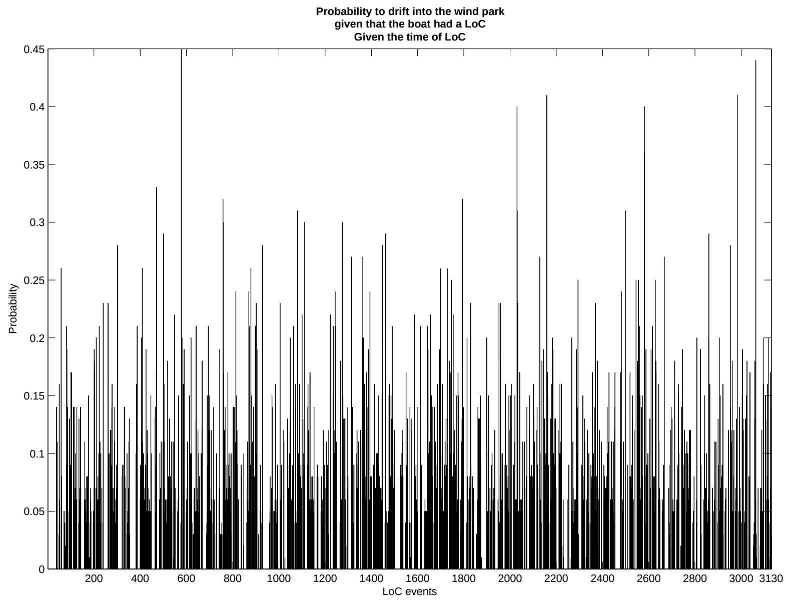

Figure 4.56: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.2.2b Collision with the whole northern part of the wind park

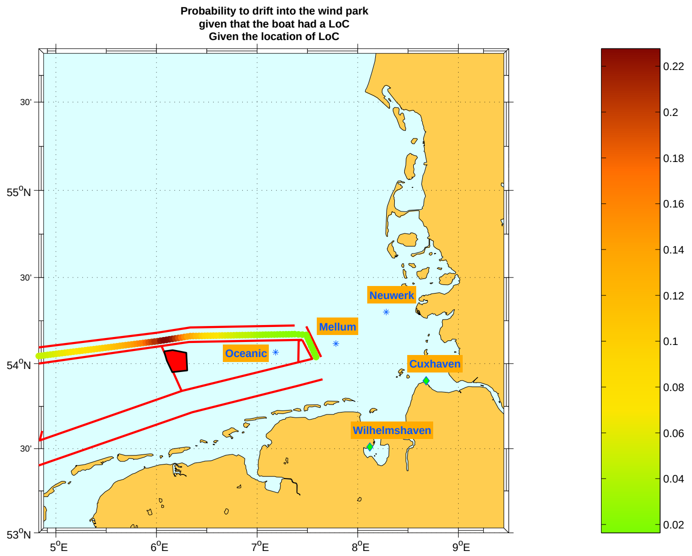

Figure 4.57: Probability to drift into the wind park given LoC and the location of the LoC

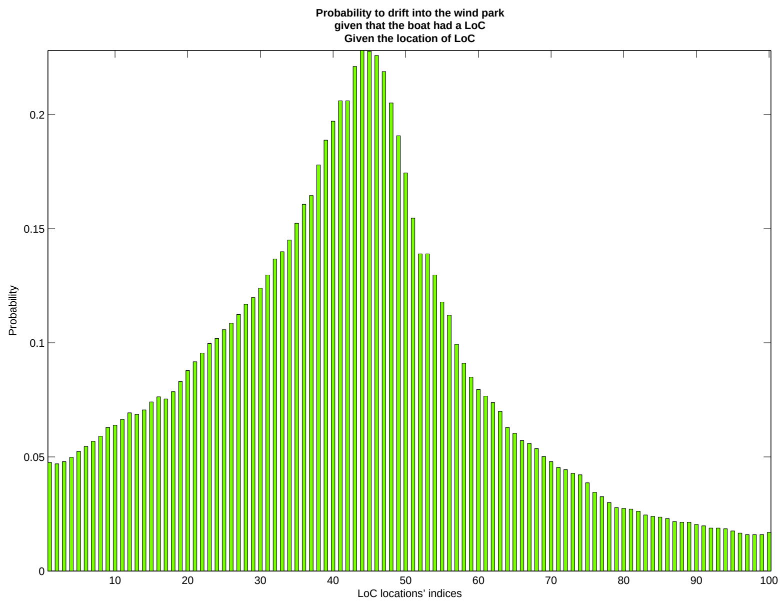

Figure 4.58: Probability to drift into the wind park given LoC and the location of the LoC

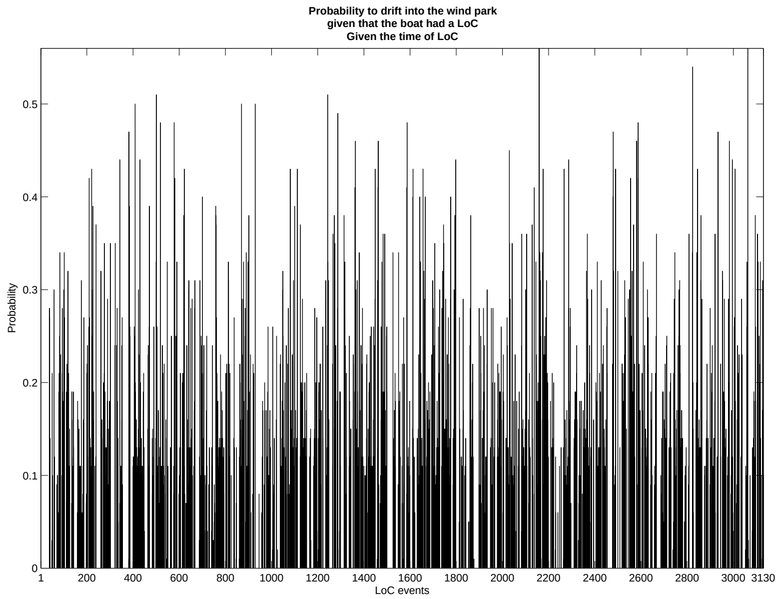

Figure 4.59: Probability to drift into the wind park given LoC and the moment of the LoC

4.3.3 Conclusions

We presented above the results of the analysis based on the database for year 1995 and also for the whole period 1990-1999. Two cases were considered. In the first case the ship was assumed to have a LoC somewhere on the northern route, and in the second case somewhere on the southern route. The probability to drift into the wind park after having a LoC can go for the northern route as high as 0.3 in 1995, depending on the location of LoC. In the case of the southern route the maximum is 0.15 in the same year. The size of the wind park is of course very important. When only the pilot section was considered for the northern route the maximum probability went down to 0.17. In all cases we found that there is a spatial concentration of the dangerous locations on the routes. The results for the whole period 1990-1999 are similar to the ones when only the year 1995 is used, just that the overall probabilities are a bit smaller this time.

Chapter 5

THE PROBABILITY OF ACCIDENT

5.1 Overview

We will define the accident as the following chain of actions:

1. A ship has a Lost of Command somewhere on its route
2. The ship starts drifting towards the wind park
3. The rescue measures fail
4. The ship collides with the wind park

We will also assume that the probability that a ship has a LoC is the same, independently of the location and the time when the ship is in a given location. Hence the probability of having an accident with the wind park can be computed using the following formula:

$$\begin{aligned} P(Acc) &= P(LoC \;\&\; Drift\ towards\ the\ wind\ park \;\&\; Towing\ failed) = \\ &= P(Towing\ failed \mid LoC \;\&\; Drift\ towards\ the\ wind\ park) * \\ &\quad * P(Drift\ towards\ the\ wind\ park \mid LoC) * P(LoC) \end{aligned}$$

Chapters three and four present two models for computing the first two probabilities in the product above. The only unknown quantity in the above formula is the probability to have a LoC. The estimated probability of having a LoC in the Wadden Sea is $1.65134221*10^{-4}$ (see "Risikoabschätzung für den Pilot park des Offshore-Wind parks Borkum Riffgrund West in Bezug auf die

Sicherheit im Seeverkehr" GAUSS mbH). The human error is also included in this probability. The probability to have a LoC was computed using the next formula $P(LoC) = \frac{mU}{A}$, where U is the number of collision-relevant LoC, $m = 5$ since it was considered that only 20% of the LoC were actually reported and $A = 37848$ is the number of ships movements in the area of interest.

5.2 Results

Based on the above formulas and probabilities, we computed the probability of collision with the wind park Borkum Riffgrund West. As in the previous analysis, we treated separately the northern and southern route. The definition of the two routes was given above. For each route we considered a different part of the wind park.

The collision probabilities obtained are shown in the next table. We will use only the results from the analysis performed using the database for the year 1995.

For the northern route we have yet another splitting based on the size and location of the wind park, namely the whole northern part of the wind park is considered or only the pilot section is considered. However, the probability to have a failed towing given that the boat had a LoC and it will drift into the wind park will not consider this separation since it will not make a big difference.

Route	P(LoC)	P(Drift to WP | LoC)	P(Towing failed | LoC & Drift to WP)	P(Collision)
Northern	$1.65*10^{-4}$	0.10	0.133	$2.23*10^{-6}$
Northern with the Pilot wind park		0.05	0.133	$1.15*10^{-6}$
Southern		0.04	0.125	$0.90*10^{-6}$

Table 5.4: The probability of collision

The expected number of collisions per year can be obtained by multiplying the probability of collision given above with the number of boats that cruise the route of interest. For example if the number of boats that pass on both routes is 40000 per year, assuming that 25% are on the northern route we have:

Route	P(Collision)	Number of boats per year	Number of collisions per year	Collision in how many years
Northern	$2.23*10^{-6}$	10000	0.0135	45
Northern with the Pilot wind park	$1.15*10^{-6}$	10000	0.0070	87
Southern	$0.90*10^{-6}$	30000	0.0138	37

Table 5.5: Estimated number of years to have a collision

As we seen the existence of a towing fleet reduces the probability of collision with approximately 90%. We will give below the probability of collision without considering any rescue measures.

First we will have to change the definition of an accident considered. An accident will be therefore defined as follows:

1. A ship has a Lost of Command somewhere on its route

2. The ship starts drifting towards the wind park

3. The ship collides with the wind park

The probability of having an accident with the wind park will be computed using the following formula:

$$P(Acc) = P(LoC \,\&\, Drift\ towards\ the\ wind\ park) =$$
$$= P(Drift\ towards\ the\ wind\ park \mid LoC) * P(LoC)$$

Now using only the results from chapter four and the probability to have a LoC that was given in chapter five the probability of accident is the following:

Route	P(LoC)	P(Drift to WP | LoC)	P(Collision)
Northern	$1.65*10^{-4}$	0.10	$1.67*10^{-5}$
Northern with the Pilot wind park		0.05	$8.65*10^{-6}$
Southern		0.04	$7.18*10^{-6}$

Table 5.6 The probability of collision

Using the same assumptions about the number of ships per each route as above we have:

Route	P(Collision)	Number of boats per year	Number of collisions per year	Collision in how many years
Northern	$1.67*10^{-5}$	10000	0.1669	6
Northern with the Pilot wind park	$8.65*10^{-6}$	10000	0.0865	12
Southern	$7.18*10^{-6}$	30000	0.2154	5

Table 5.7 Estimated number of years to have a collision

Chapter 6

CONCLUSIONS

Risk assessment in maritime environment is a very complicated issue that requires multiple analyses to be performed in order to cover all the aspects. There are two constraints that govern a risk assessment project for the maritime environment:

1. There is not sufficient data in order to create a complete logical system containing all possible causal chains in a very busy port or shipping route.

2. Data describing human error and other basic failures are either not available or if available are partial, misleading or just poorly documented.

One big advantage of this study was the existence of a very accurate weather database for the region under analysis.

This study presented three models that were combined in order to compute the probability of having a collision with the offshore wind parks which will be installed in the near future adjacent to shipping areas in the German Bight that are among the busiest in the world.

Different scenarios were studied for each of the model, depending on the route on which the boat that caused the accident was cruising and also on the place of the accident.

The emergency towing operations to avert a collision of a vessel in distress with a wind turbine were also included in the analysis. Hence the probability to have a successful emergency towing was computed. Different towboats were considered. The analysis showed that having a towing strategy implemented is vital in order to maintain a risk free maritime environment.

Matlab programs were developed to implement the models. The analyses covered the period January 1990 to December 1999.

The probability to have an accident is quite large. This is better shown by the expected number of years to have a collision. When the towing fleet is present we expect an accident in less than 90 years. However if no rescue measure is implemented we expect an accident in less than 15 years. In the GAUSS' report, introduced above, the expected number of years to have a collision with the wind park is in the order of thousands. Hence there is a huge difference with the results above. Because the weather data is not available, the analysts from GAUSS are forced to make assumptions. Actually this is the case in most studies. Hence they assume a fixed number of possible wind directions. The next assumption is that the wind direction does not change on the entire drifting period. This is an extremely bad assumption since the wind direction can change extremely fast. We can see that in the Figure 3.40 from page 43. Without having a ship drift model in GAUSS' study it is assumed that the drifting trajectory will always be inside a 45° sector. In the present study a ship drift model was developed in order to compute the trajectory of a drifting boat based on the weather conditions.

Hence having an accurate weather database can completely change the picture of the risk in the domain under study. Of course the results of the present study are also extremely dependent on the accuracy of the ship drifting model. Thus a recommendation for future research will be to improve the drifting model.

An analysis of costs and "best environmental benefits" might be the next step to complete this study. In the diagram from page 9 this is coded in yellow. This type of analysis might consider the economical lost and the damages made on the ecosystem of the region. For this a good understanding of the ecosystem is required, such as what is the composition of the fauna in every moment of the year. Before starting an analysis of the consequences of an oil spill on the fauna of the region is required first to answer to questions like: Are there species that are in danger of extension? Is there any special treatment for different species?

The results from the other side of this project, which analyzes the consequences of oil spill on the environment implementing also a strategy for mechanical cleaning, should be corroborated with the results from this report in order to have a better image of the risk. An often used definition of risk is the product of the probability to have the event, in this case an oil spill, with the consequences resulted.

BIBLIOGRAPHY

[1] Ole Morten Aamo, Mark Reed and Keith Downing, *"Oil spill contingency and response (OSCAR) Model system: Sensitivity studies"*, Paper id #088, International Oil Spill Conference, 1997.

[2] Committee on Risk Assessment and Management of Marine Systems, *"Review of the Prince William Sound, Alaska, Risk Assessment Study"* National Academy Press, 1998.

[3] Jason R.W. Merrick, J. Rene van Dorp, Jack Harrald, Thomas Mazzuchi, John E. Spahn and Martha Grabowski, *"A systems Approach to Managing Oil Transportation Risk in Prince William Sound",* System Engineering, Vol. 3, No. 3, 2000

[4] Huw Jones, Dr. Luke Chippindall, Dr. Nick Hardy, *"Risk Assessment of pollution from spills in Australian Waters',* SPILLCON 2000, Darwin, 15 August 2000

[5] J. Rene van Dorp, Jason R.W. Merrick, Jack R. Harrald, Thomas Mazzuchi and Martha Grabowski, *"A Risk Management Procedure for the Washington State Ferries",* Risk Analysis, Vol. 21, No. 1, 2001

[6] Jason R.W. Merrick, J. Rene van Dorp, Jack Harrald, Thomas Mazzuchi, John E. Spahn and Martha Grabowski, *"The Prince William Sound Risk Assessment",* INFORMS Vol. 32, No. 6, November-December 2002, pp. 25-40

[7] Jason R.W. Merrick, J. Rene van Dorp, Joseph P. Blackford, Gregory L. Shaw, Jack Harrald, Thomas A. Mazzuchi, *"A traffic density analysis of proposed ferry service expansion in San Francisco Bay using a maritime simulation model"* Reliability Engineering and System Safety 81, 2003, pp. 119-132

[8] Ole Morten Aamo, Mark Reed and Alun Lewis, *"Regional Contingency Planning Using OSCAR Oil Spill Contingency and Response Model",* SINTEF Applied Chemistry, Environmental Engineering, N-7034 Trondheim, Norway

[9] Carlos Guedes Soares, Ralf Weisse, Juan Carlos Carretere and Enrique Alvarez, *"A 40 years hindcast of wind, sea level and waves in European waters",* 21st International Conference on Offshore Mechanics and Artic Engineering, 23-28 June 2002 in Oslo, Norway

APPENDIX A

In this section we will present the Matlab code used to perform the analyses. For each of the two analyses we will list the main functions. The comments will be included trough the code and will be introduced using Matlab syntax, namely using the symbol "%".

Towing simulation

There are two main functions used for the computation of the probability to tow a drifting boat. The function "Towing" is performing the actual simulation of the towing mission. It records information like the outcome of the mission, the time needed to save the boat and the location of the boat. Next, the function "extractstatistics" will compute offline the statistics.

1. Function: Towing

```
function res = Towing(eventid, varargin)
% For a given event compute the probability of being able to tow the
drifting boat before having an accident

% Define the towing constants
towingconstants;

% Define the plotting function if plot is needed the figure to plot on
should be already created
if(TowingNeedPlot)
    % Define the ploting system constants
    plotingconstants;
    % Take the ploting function
    plotFun = str2num(PlotFun);
    % Take the text function
    textFun = str2num(TextFun);
    hold on;
    towingPlotHandlers = [];
end

% Define the constants needed to work on the earth surface
ellipsoidconstants;
ellipsoidDistFun = str2num(EllipsoidDistFun);

% We need to know everything about towboats
loadTowboatsDB;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%  Load the trajectory data  %%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Example of eventid: 19981107 or 1995110716
% The data file is located in the directory:
```

```
% ../Data/subdir/SHIP_DRIFT_TRAJECTORIES/YEAR_1998/MONTH_11
% The name of the file is: ship_drift_19981107.dat
% or ship_drift_1998110716.dat this only if the time between
% events is not 24 hours

% Find if there is an additional subdir
subdir = Subdir;

% Make it a string
eventidstr = num2str(eventid);

% Find the year of the accident
year = eventidstr(1:4);

% Find the month of the accident
month = eventidstr(5:6);

% Get the path and the name of the data file
datapath = strcat(['../Data/Trajectories/', AccidentType, '/', subdir,
'/SHIP_DRIFT_TRAJECTORIES/YEAR_', ...
    year, '/MONTH_', month]);
datafile = strcat(['ship_drift_', eventidstr, '.dat']);

fid = fopen(strcat([datapath, '/', datafile]));
if(fid == -1)
    res.error = -1; % The file was not found
    return
else
    aux = 0;
    while(aux == 0)
        % Get the next line
        nextline = fgetl(fid);
        % Check if all the text from the beginning of the file was skiped
        if(length(str2num(nextline))~=0)
            aux = 1;
        end;
    end
    res.error = 0; % The file was found
    % Load the trajectories
    %                    ind lon/lat x/y wd time
    data = fscanf(fid, '%d %f %f %f %f %f %lg', [7, inf]);
    data = data';
end

% Close the file
fclose(fid);

% Split the data in:
%       -- backward trajectory and
%       -- forward trajectory
% Get backward trajectory
index = find(data(:,1)<=0);
bw = data(index,:);

% Get the forward trajectory
index = find(data(:,1)>0);
fw = data(index,:);

%Get the accident
index = find(data(:,1)==0);
acc = data(index,:);

% Delete unnecessary variables
clear data

% If there is no backward data meaning that the backward trajectory never
crosses the routes return here
if(size(bw, 1) == 1)
    res.error = 2;
```

```
    return
end

% Update the number of hours that the trajectories were followed
LengthBackwardTrajectory = size(bw, 1) - 1;
LengthForwardTrajectory = size(fw, 1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%  Load the wind and wave data  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% We suppose that we have already loaded the data on the wind and waves.
The data is given hourly in the interval [StartingDateTime, EndingDateTime].
The wind data is given at a specified location and contains only the wind
speed in knots. The wave data is given at a specified location and contains
only the wave height in meters loadwindwavedata;

% Get the local weather condition make the eventid a string
eventidstr = num2str(eventid);
if(EventTimeInterval == 24)
    currEvent = datevec(addtodate(datenum(strcat([eventidstr(5:6),'/',...
        eventidstr(7:8), '/',...
        eventidstr(1:4)])) + .0625, -
                                 LengthBackwardTrajectory/24, 'day'));
    % Set the hour
    currEvent(4) = 0;
    % Clean the mess
    currEvent = currEvent(1:4);

    startint = currEvent*[1e6 1e4 1e2 1]';

    currEvent = datevec(addtodate(datenum(strcat(
[eventidstr(5:6),'/', eventidstr(7:8), '/',...
        eventidstr(1:4)])) + .0625,
LengthForwardTrajectory/24, 'day'));
    % Set the hour
    currEvent(4) = 0;
    % Clean the mess
    currEvent = currEvent(1:4);
    endint = currEvent*[1e6 1e4 1e2 1]';
else
    hh = abs(mod(str2double(eventidstr(9:10)) –
LengthBackwardTrajectory, 24));
    transfer = floor((str2double(eventidstr(9:10)) –
            LengthBackwardTrajectory)/24);
    currEvent = datevec(addtodate(datenum(strcat(
[eventidstr(5:6),'/', eventidstr(7:8), '/',...
        eventidstr(1:4)])) + .0625, transfer, 'day'));
    % Set the hour
    currEvent(4) = hh;
    % Clean the mess
    currEvent = currEvent(1:4);
    startint = currEvent*[1e6 1e4 1e2 1]';

    hh = abs(mod(str2double(eventidstr(9:10)) +
              LengthForwardTrajectory, 24));
    transfer = floor((str2double(eventidstr(9:10)) +
LengthForwardTrajectory)/24);
    currEvent = datevec(addtodate(datenum(strcat(
[eventidstr(5:6),'/', eventidstr(7:8), '/',...
        eventidstr(1:4)])) + .0625, transfer, 'day'));
    % Set the hour
    currEvent(4) = hh;
    % Clean the mess
    currEvent = currEvent(1:4);
    endint = currEvent*[1e6 1e4 1e2 1]';
```

```
end
localwinddata = winddata(find(winddata(:,1) >= startint &
winddata(:,1) <= endint),:);
localwavedata = wavedata(find(wavedata(:,1) >= startint &
wavedata(:,1) <= endint),:);

% Init statistics
res.outcome = zeros(NumTrials, 1); % 1 is any towboat has
succeeded to rescue the drifting boat in time, 0 otherwise
for i = 1:NumTrials
    for j = 1:towboats.noTowboats
res.boat{i,j} = ''; % '' if no towboat has succeded, otherwise the
name of the rescuer
    end
end;
res.loc = zeros(NumTrials, 2*towboats.noTowboats); % the location
of the boat when rescued
res.time = Inf*ones(NumTrials, towboats.noTowboats); % number of
hours from the begining of drifting
% Add this to shorten the execution time. Generate the random
positions of the boats.
for currTowboat = 1:towboats.noTowboats
    if(~towboats.boats{currTowboat}.fixLocation)
        randomLocation{currTowboat} = mgd(NumTrials + 2, 2,...
            towboats.boats{currTowboat}.locationMean',...
            towboats.boats{currTowboat}.locationCovariance);
    end
end

% Add to speed up process and correctly generate the random
numbers
ATrand = rand(NumTrials*towboats.noTowboats,1);
ATindex = 0;
RTrand = rand(NumTrials*towboats.noTowboats,1);
RTindex = 0;
CTrand = rand(NumTrials*towboats.noTowboats*10,1);
CTindex = 0;
CTstate = rand('state');
FSTrand = rand(NumTrials*towboats.noTowboats*10,1);
FSTindex = 0;
FSTstate = rand('state');

%%%%%%%%%%%%%%%%%%%%%%%%%
% START THE SIMULATION %
%%%%%%%%%%%%%%%%%%%%%%%%%
for currTrial = 1:NumTrials
    %----------------%
    % Initialization %
    %----------------%
    trialres.outcome = zeros(towboats.noTowboats, 1); % 1 is any towboat
has succeded to rescue the drifting boat in time, 0 otherwise
    trialres.loc = zeros(towboats.noTowboats, 2); % the location of the
boat when rescued
    trialres.time = Inf*ones(towboats.noTowboats, 1); % number of hours
from the begining of drifting

    for currTowboat = 1:towboats.noTowboats
        % Clean the plot before
        if(TowingNeedPlot)
            if(~isempty(towingPlotHandlers))
                pause(2);
                % Delete the last plots of the routes
                for i = 1:length(towingPlotHandlers)
                    delete(towingPlotHandlers(i));
                end
```

```
                towingPlotHandlers = [];
            end
        end
        %------------------------%
        % Get the current towboat %
        %------------------------%
        towboat = towboats.boats{currTowboat};
        %---------------------------------%
        % Simulate the alarm time in hours %
        %---------------------------------%
        ATindex = ATindex + 1;
        alarmTime = towboat.alarmTime(1,1) +
(towboat.alarmTime(1,2) -
towboat.alarmTime(1,1))*ATrand(ATindex);

        % askIfNeedUnloading is an auxiliary variable that does exactly
what it says :)
        askIfNeedUnloading = 0;
        %--------------------------%
        % Find the towboat location %
        %--------------------------%
        if(towboat.fixLocation)
            boatLocation = towboat.location;
        else
% If the towboat does not have a fix location by its nature it could have a
fix location if the weather is worst that some limit at the time of
beginning of drifting.
            if(localwinddata(1,2) >=
towboat.fixLocationTrigger)
                boatLocation = towboat.location;
            else
                boatLocation =
                        (randomLocation{currTowboat}(currTrial,:))';

% Does the towboat need to go to its harbor to unload?
% If yes then find the time to go there and place the boat in its harbour.
                askIfNeedUnloading = 1;
            end;
        end
% If needed plot the boat position
        if(TowingNeedPlot)
            pause(0.5);
            if(LatLon)
                towingPlotHandlers        =        plotFun(boatLocation(1),
boatLocation(2), 'r*', 'MarkerSize', 3);
                towingPlotHandlers(end+1) =
textFun(boatLocation(1)  +  1/30,  boatLocation(2)  +  1/30,  towboat.name,
'FontWeight', 'demi');
            else
                towingPlotHandlers        =        plotFun(boatLocation(2),
boatLocation(1), 'r*', 'MarkerSize', 3);
                towingPlotHandlers(end+1) =
textFun(boatLocation(2)  +  1/30,  boatLocation(1)  +  1/30,  towboat.name,
'FontWeight', 'demi');
            end
        end

        %----------------------%
        % Get the towboat speed %
        %----------------------%
        % May be altered be weather
        towboatSpeed = towboat.speed;

        %-----------------------------------%
        % Simulate the time to react in hours %
        %-----------------------------------%
        RTindex = RTindex + 1;
        reactTime =  towboat.reactTime(1,1) +
(towboat.reactTime(1,2) -
```

```
towboat.reactTime(1,1))*RTrand(RTindex);

        %-------------------------------------%
        % Check if the towboat needs to unload %
        %-------------------------------------%
        % The time to go to the harbor
        goToHarbourTime = 0;
        % Does the towboat need to go to its harbor to unload?
        % If yes then find the time to go there and place the boat in its
harbour.
        if( askIfNeedUnloading )
            if(rand <= towboat.probNeedUnloading)
                % If needed plot the boat position
                if(TowingNeedPlot)
                    pause(0.5);
                    if(LatLon)
                        towingPlotHandlers(end+1) =
plotFun([boatLocation(1); towboat.baseLocation(1)],
[boatLocation(2); towboat.baseLocation(2)], 'c-');
                        towingPlotHandlers(end+1) =
plotFun(towboat.baseLocation(1),       towboat.baseLocation(2),       'c*',
'MarkerSize', 3);
                    else
                        towingPlotHandlers(end+1) =
plotFun([boatLocation(2); towboat.baseLocation(2)],
[boatLocation(1); towboat.baseLocation(1)], 'c-');
                        towingPlotHandlers(end+1) =
plotFun(towboat.baseLocation(2),       towboat.baseLocation(1),       'c*',
'MarkerSize', 3);
                    end
                end

                goToHarbourTime =
ellipsoidDistFun( boatLocation(1), boatLocation(2),...
                    towboat.baseLocation(1),
towboat.baseLocation(2))/towboatSpeed;
                boatLocation = towboat.baseLocation;
            end
        end

        % Add the time needed to go to its harbor. If the towboat does not
need to go to its harbor this will be 0.
        reactTime = reactTime + goToHarbourTime;

        %------------------------------------%
        % Find the intersection time in hours %
        %------------------------------------%
        % Get the starting time index. The time that passed until the
towboat is ready to go expressed in number of time steps
        startTimeIndex = ceil((alarmTime +
reactTime)/DriftTimeInterval) + 1;
        % Get the time needed for the drifting vessel to go to the first
possible meeting place
        residualTime = ceil((alarmTime +
reactTime)/DriftTimeInterval)*DriftTimeInterval - (alarmTime + reactTime);

        % Init the dist
        dist = zeros(size(bw, 1), 1);

        % Is there any chance to continue?
        if(startTimeIndex >= length(dist))
            % If not then go to the next towboat
            continue;
        end;

        howmany = ones(length(dist)-startTimeIndex+1,1);
        % Compute the distance between the towing boat and the drifting
        dist(startTimeIndex:end) =
ellipsoidDistFun(boatLocation(1)*howmany, boatLocation(2)*howmany, ...
```

```
                bw(startTimeIndex:length(dist),3),
bw(startTimeIndex:length(dist),2));
        % Get the time needed: time = dist/speed in hours and add 10% to
the time needed to intercept the boat - GKSS
        dist = 1.1*dist/towboatSpeed;

        % Compute the time difference
        dist(startTimeIndex:end) = ((0:(length(dist) -
startTimeIndex))'*DriftTimeInterval + residualTime) -
dist(startTimeIndex:end);
        % Check if the towboat and the drifting boat are intersecting. The
towboat should arrive at the crossing position maximum 1 minutes = 1/60
hours later than the drifting boat
        index = find(dist(startTimeIndex:end) >= -1/60);

        % Is there any chance to continue?
        if(isempty(index))
        % The boat cannot be rescued go to the next towboat
            continue;
        end

        % There are times were the boat can be rescued. Make index be an
index in the entire database
        index = index + startTimeIndex - 1;

        % If any of the possible meeting point is in shallow waters then
remove it from the database 2m more
        index = index(bw(index,6) > towboat.draught + 2);
        % Is there any chance to continue?
        if(isempty(index))
        % The boat cannot be rescued go to the next towboat
            continue;
        end

        % Choose only the first possible meeting point
        % ALWAYS USE ONLY YHE FIRST MEETING POINT - GKSS
        index = index(1);
      % For each place were the boats were close to each other try to see
if the action would succeeded. The time was updated; hence we are now at
the connection location
        for i=1:length(index)
            % Get the place where the two boats are meeting
            location = [bw(index(i), 3), bw(index(i), 2)];

            % If needed plot the boat position
            if(TowingNeedPlot)
                pause(0.5);
                if(LatLon)
                    towingPlotHandlers(end+1) =
plotFun([boatLocation(1); location(1)],
[boatLocation(2);location(2)], 'b-');
                    towingPlotHandlers(end+1) =
plotFun(location(1), location(2), 'b*', 'MarkerSize', 3);
                else
                    towingPlotHandlers(end+1) =
plotFun([boatLocation(2); location(2)],
[boatLocation(1);location(1)], 'b-');
                    towingPlotHandlers(end+1) =
plotFun(location(2), location(1), 'b*', 'MarkerSize', 3);
                end
            end

%--------------------------------------------------------%
% Simulate the time needed to connect the vessels in hours %
%--------------------------------------------------------%                %
maybe based on the weather at the location
            CTindex = CTindex + 1;
            if(CTindex > length(CTrand))
                rand('state', CTstate);
```

```
                CTrand = rand((NumTrials -
currTrial+1)*towboats.noTowboats*10, 1);
                CTindex = 1;
                CTstate = rand('state');
            end

            connectTime = towboat.connectingTime(1,1) +
(towboat.connectingTime(1,2)      -      towboat.connectingTime(1,1))      *
CTrand(CTindex);

            % Get the new location if there is one
            if(round(connectTime/DriftTimeInterval) + index(i)
>= length(dist))
            % During connection the boat had the accident
            % Go to the next point of intersection
                continue;
            end;

            % Get the place where the two boats are linked
            location = [bw(round(connectTime/DriftTimeInterval) + index(i),
3),...
                bw(round(connectTime/DriftTimeInterval) +
index(i), 2)];
            % If needed plot the boat position
            if(TowingNeedPlot)
                pause(0.5);
                if(LatLon)
                    towingPlotHandlers(end+1) =
plotFun(location(1), location(2), 'm*', 'MarkerSize', 3);
                else
                    towingPlotHandlers(end+1) =
plotFun(location(2), location(1), 'm*', 'MarkerSize', 3);
                end
            end

            %-----------------------------------------------%
            % Simulate the time needed to full stop in hours %
            %-----------------------------------------------%
            % maybe based on the weather at the location
            FSTindex = FSTindex + 1;
            if(FSTindex > length(FSTrand))
                rand('state', FSTstate);
                FSTrand = rand((NumTrials -
currTrial+1)*towboats.noTowboats*10, 1);
                FSTindex = 1;
                FSTstate = rand('state');
            end
             fullStopTime = towboat.fullStopTime(1,1) +
(towboat.fullStopTime(1,2) -
towboat.fullStopTime(1,1))*FSTrand(FSTindex);

            % Check if the boat was saved or not
            if(round((fullStopTime +
connectTime)/DriftTimeInterval) + index(i) >= length(dist))
                % During stopping the boat had the accident
                % Go to the next point of intersection
                continue;
            end;

            % If  we  are  here  it  means  that  the  boat  was  saved  by  this
towboat
            trialres.outcome(currTowboat) = 1;
            % Is this faster then the other
            if(trialres.time(currTowboat) > (fullStopTime +
connectTime + index(i)*DriftTimeInterval))
                trialres.time(currTowboat) = fullStopTime +
connectTime + index(i)*DriftTimeInterval;
                trialres.loc(currTowboat,:) =
```

```
[bw(round(trialres.time(currTowboat)/DriftTimeInterval),            3),...
bw(round(trialres.time(currTowboat)/DriftTimeInterval), 2)];
                % If needed plot the boat position
                if(TowingNeedPlot)
                    pause(0.5);
              % Get the place where the two boats are linked
                    if(LatLon)
                        towingPlotHandlers(end+1) =
plotFun(trialres.loc(currTowboat,1), trialres.loc(currTowboat,2), 'y*');
                        towingPlotHandlers(end+1) =
textFun(trialres.loc(currTowboat,1) + 1/30,
trialres.loc(currTowboat,2)   +   1/30,   'SUCCESS!!!',   'Color',   'b',
'FontWeight', 'demi');
                    else
                        towingPlotHandlers(end+1) =
plotFun(trialres.loc(currTowboat,2), trialres.loc(currTowboat,1), 'y*');
                        towingPlotHandlers(end+1) =
textFun(trialres.loc(currTowboat,2) + 1/30,
trialres.loc(currTowboat,1)   +   1/30,   'SUCCESS!!!',   'Color',   'b',
'FontWeight', 'demi');
                    end
                end
             end
        end % for each intersection place
    end % for each towboat

    % Get the result of the trial
    if (any(trialres.outcome))
        res.outcome(currTrial) = 1; % 1 is any towboat has succeded to
rescue the drifting boat in time, 0 otherwise
        [aux, index] = sort(trialres.time); % Sort the time

        for i = 1:sum(trialres.outcome)
            res.boat{currTrial, i} =
towboats.boats{index(i)}.name;  %  'none'  if  no  towboat  has  succeded,
otherwise the name of the rescuer
            res.loc(currTrial, 2*i-1:2*i) =
trialres.loc(index(i), :); % the location of the boat when rescued
            res.time(currTrial, i) = trialres.time(index(i)); % number of
hours from the begining of drifting
        end
    end
end % for each trial

% Modify result so that only the number of successes are remembered
res.outcome = sum(res.outcome);
```

2. Function: extractstatistics

```
% This function will extract statistics from the results
function stat = extractstatistics(isfilename, data)

% The result can be in a .mat file and the name of the file is provided.
if (isfilename)
    load(strcat(['Results/', data, '.mat']));
    data = res;
end

% Get the towboats database
loadTowboatsDB;

% Get the towing constants
towingconstants;
```

```
% Get the names of the towboats
for i=1:towboats.noTowboats
    stat.towboatsNames{i} = towboats.boats{i}.name;
end;

% Get the mean and variance for time and place of the rescue
% per all events.
stat.totalProbTowboat = [ 0 0 0];
stat.totalTimeMean = [];
stat.totalTimeVariance = [];
stat.totalLocationMean = [];
stat.totalLocationVariance = [];

stat.numEvents = 0;
stat.impossibleEvents = [];

% Probability of having an accident only from the point of view of towing
probab = [];
probab_index = 0;

for currEvent = 1:length(data)

    % Check if  the event exists
    if(data{currEvent}.result.error ~= -1)

        % Next event
        stat.numEvents = stat.numEvents + 1;

        % Get the eventid
        stat.event{stat.numEvents}.eventId = data{currEvent}.eventId;

        % Next event
        probab_index = probab_index + 1;

        % Check if this event is possible
        if(data{currEvent}.result.error == 2) % This event is not possible
            stat.impossibleEvents          =          [stat.impossibleEvents
data{currEvent}.eventId];
            stat.numEvents = stat.numEvents - 1;

            % This accident has probability 0 to occur
            probab(probab_index) = 0;

            continue;
        end

        % Get the relative frequency of rescue success
        stat.event{stat.numEvents}.succProb                               =
data{currEvent}.result.outcome/NumTrials;

        % This event has probability 1 - succProb to occur
        probab(probab_index) = 1 - stat.event{stat.numEvents}.succProb;

        % Get the probability of being rescued by a given towboat
        % This probability is conditioned by the success of the rescue
        stat.event{stat.numEvents}.towboatprob          =          zeros(1,
length(stat.towboatsNames));
        if(data{1,currEvent}.result.outcome > 0)
            for currTowboat=1:length(stat.towboatsNames)
                currTowboatName = stat.towboatsNames{currTowboat};
                stat.event{stat.numEvents}.towboatprob(currTowboat)       =
length(strmatch(currTowboatName, data{1,currEvent}.result.boat(:,1)))...
                    /data{1,currEvent}.result.outcome;
            end;
        end

        % Compute the overall probability to be rescued by a given towboat
        % conditioned by the fact that the rescue mission had succedded
```

```
        stat.totalProbTowboat       =       stat.totalProbTowboat       +
stat.event{stat.numEvents}.towboatprob;

        % Get the mean and the variance of the location
        % conditioned by the fact that the rescue mission had succedded
        currEventLoc = data{1,currEvent}.result.loc(:,1:2);
        % Eliminate the 0 values;
        currEventLoc = currEventLoc(find(currEventLoc(:,1) ~= 0), :);
        if(size(currEventLoc, 1)>0)
            % Compute the mean location
            stat.event{stat.numEvents}.locationMean = mean(currEventLoc);
            % Compute the variance of the location
            stat.event{stat.numEvents}.locationVariance = var(currEventLoc);
        else
            stat.event{stat.numEvents}.locationMean = [NaN, NaN];
            stat.event{stat.numEvents}.locationVariance = [NaN, NaN];
        end

        % Update for the total location statistics
        % This is just a temporary storage
        stat.totalLocationVariance      =      [stat.totalLocationVariance;
currEventLoc];

        % Get the mean and the variance of the time
        currEventTime = data{1,currEvent}.result.time(:,1);
        % Eliminate the Inf values;
        currEventTime = currEventTime(find(currEventTime(:,1) ~= Inf),1);
        if(length(currEventTime)>0)
            % Compute the mean location
            stat.event{stat.numEvents}.timeMean = mean(currEventTime);
            % Compute the variance of the location
            stat.event{stat.numEvents}.timeVariance = var(currEventTime);
        else
            stat.event{stat.numEvents}.timeMean = NaN;
            stat.event{stat.numEvents}.timeVariance = NaN;
        end

        % Update for the total time statistics
        % This is just a temporary storage
        stat.totalTimeVariance = [stat.totalTimeVariance; currEventTime];
    end
end

% Compute the overall probability to be rescued by a given towboat
% conditioned by the fact that the rescue mission had succedded
stat.totalProbTowboat = stat.totalProbTowboat/stat.numEvents;

% Get the total time mean and variance
stat.totalTimeMean = mean(stat.totalTimeVariance);
stat.totalTimeVariance  = var(stat.totalTimeVariance);

% Get the total loaction mean and variance
stat.totalLocationMean = mean(stat.totalLocationVariance);
stat.totalLocationVariance = var(stat.totalLocationVariance);

% Make a nice figure
figure;
hold on
axis off
%title(sprintf('Statistics\nAccident Type: %s', strcat([AccidentType, ' 1st
meeting point'])), 'FontSize', 12);
title(sprintf('Statistics\nAccident  Type:  %s',  AccidentType),  'FontSize',
12);
h = pie3(stat.totalProbTowboat, [1 1 1], stat.towboatsNames);
h = pie3(stat.totalProbTowboat, [1 1 1]);
%rotate(h, [1 0 0], 65);
%rotate(h, [0, 1, 0], 50);
axis tight
```

```
% Make a second nice picture
figure;
bar(probab);
hold on
%title(sprintf('Statistics\nAccident Type: %s', strcat([AccidentType, ' 1st meeting point'])), 'FontSize', 12);
title(sprintf('Statistics\nAccident Type: %s', AccidentType), 'FontSize', 12);
xlabel('Time of the event');
set(gca, 'XTickLabelMode', 'manual');
set(gca, 'XTickMode', 'manual');
set(gca, 'XTick', [1,length(probab)]);
%set(gca, 'XTickLabel', {'January 1st, 1995', 'December 31st, 1995'});
ypos = get(gca,'Ylim');
ypos = ones(2, 1)*ypos(1)*0.996;
xpos = str2num(get(gca,'XTickLabel'));
xpos = xpos([1,end]);
text(xpos, ypos, {'January 1st, 1995', 'December 31st, 1995'}, 'Rotation', 5)
ylabel('Probability to have an accident in this event');
set(gca, 'XTickLabel', {'',''});
%plot([1:length(probab)], probab);

axis tight;
hold off

% Save nice file
answer = input('Do you want a nice file?[y/n]  ', 's');
while length(strmatch(answer, {'y', 'n'})) == 0
    answer = input('Please answer only with y or n  ', 's');
end

if(answer == 'y')
    answer = input('Please input the name of the file ', 's');

    % Set the path and the name of the data file
    pathnamefile = strcat(['Results/', answer, '.dat']);
    fid = fopen(pathnamefile);
    while (fid ~= -1)
        fclose(fid);
        answer = input('The file that you specified already exists!!!\nPress <Enter> to overwrite or input new file name ', 's');
        if(length(answer))
            pathnamefile = strcat(['Results/', answer, '.dat']);
        else
            break;
        end
        fid = fopen(pathnamefile);
    end

    % Open the file for reading
    fid = fopen(pathnamefile, 'w');
    if(fid == -1)
        disp('Unexpected error!!! Exit now');
        return % The file was not found
    else
        fprintf(fid, 'The accident is assumed to be caused by a vessel coming from %s Route(s)\n', AccidentType);
        fprintf(fid, 'Towboats: The assumptions for the towboats are listed in the towboatsDB.dat file\n');
        fprintf(fid, 'Number of possible events: %d\nNumber of trials per event: %d\n', stat.numEvents, NumTrials);
        fprintf(fid, 'Number of impossible events: %d\n', length(stat.impossibleEvents));
        fprintf(fid, 'List of impossible events:\n');
        fprintf(fid, ' %d, %d, %d, %d\n', stat.impossibleEvents);
        fprintf(fid, '\n');
        fprintf(fid, 'Date of the first event: %s\n', datestr(datenum([StartingDateTime 00 00])));
```

```
        fprintf(fid,     'Date     of     the     last     event:     %s\n',
datestr(datenum([EndingDateTime 00 00])));
        fprintf(fid, '===============================\n\n');
        fprintf(fid, '==================\n');
        fprintf(fid, '= Overall results =\n');
        fprintf(fid, '==================\n\n');
        s = 0;
        for i=1:stat.numEvents
            s = s + stat.event{1,i}.succProb;
        end
        fprintf(fid,  'The  overall  probability  to  be  rescued:  %f\n',
s/stat.numEvents);
        fprintf(fid, 'The overall probability to be rescued by each towboat
given that the mission was successfull\n         ');
        for i=1:length(stat.totalProbTowboat)
            fprintf(fid,     '%s     :     %f         ',stat.towboatsNames{i},
stat.totalProbTowboat(i));
        end
        fprintf(fid, '\n');
        fprintf(fid,   'The   overall   mean   time   until   rescued   %f\n',
stat.totalTimeMean);
        fprintf(fid,  'The  overall  variance  of  time  until  rescued  %f\n',
stat.totalTimeVariance);
        fprintf(fid,  'The  overall  mean  location  when  rescued  [%f  %f]\n',
stat.totalLocationMean);
        fprintf(fid, 'The overall variance of the location when rescued [%f
%f]\n', stat.totalLocationVariance);
        fprintf(fid, '===============================\n\n');

        fprintf(fid, '====================\n');
        fprintf(fid, '= Results per event =\n');
        fprintf(fid, '====================\n\n');
        for currEvent = 1:stat.numEvents
            fprintf(fid, 'EventId %d\n', stat.event{currEvent}.eventId);
            fprintf(fid,     'The     probability     of     success     %f\n',
stat.event{1,currEvent}.succProb);
            fprintf(fid,  'The  probability  to  be  rescued  by  each  towboat
given that the mission was successfull\n         ');
            for i=1:length(stat.totalProbTowboat)
                fprintf(fid,    '%s    :    %f        ',stat.towboatsNames{i},
stat.event{currEvent}.towboatprob(i));
            end
            fprintf(fid, '\n');
            fprintf(fid,    'The    mean    time    until    rescued    %f\n',
stat.event{currEvent}.timeMean);
            fprintf(fid,   'The   variance   of   time   until   rescued   %f\n',
stat.event{currEvent}.timeVariance);
            fprintf(fid,   'The   mean   location   when   rescued   [%f   %f]\n',
stat.event{currEvent}.locationMean);
            fprintf(fid,  'The  variance  of  the  location  when  rescued  [%f
%f]\n', stat.event{currEvent}.locationVariance);
            fprintf(fid, '--------------------------------------\n\n');
        end

        % Close the file
        fclose(fid);
    end
end
```

Collision probability

The following function computes the probability to drift into the wind park given that the boat had a LoC and no mitigation measures are taken. The results of this analysis are presented in chapter 5.

1. Function colprobab

```
function [p1, p2, p3] = colprobab
% Define the global parameters
params;

% Find if there is an additional subdir
subdir = Subdir;

% Generate the events
eventIndex = 0;
lastEventProcessed = 0;
currEvent = StartingDateTime;
while(~lastEventProcessed)

    eventId = currEvent(1)*1000000 + currEvent(2)*10000 + currEvent(3)*100
+ currEvent(4);

    % Next event
    eventIndex = eventIndex + 1;

    % Make it a string
    eventidstr = num2str(eventId);

    % Find the year of the accident
    year = eventidstr(1:4);

    % Find the month of the accident
    month = eventidstr(5:6);

    % Get the path and the name of the data file
    datapath      =      strcat(['../Data/OtherTrajectories/',      subdir,
'/SHIP_DRIFT_TRAJECTORIES/YEAR_', year, '/MONTH_', month]);

    % Read all data related with this event
    for currPt = 1:NoPoints

        datafile = strcat(['ship_drift_', eventidstr, '_', sprintf('%03d',
                   currPt), '.dat']);
        % Open the data file
        fid = fopen(strcat([datapath, '/', datafile]));
        if(fid == -1)
            error('Event file not found'); % The file was not found
            return
        else
            aux = 0;
            while(aux == 0)
                % Get the next line
                nextline = fgetl(fid);
                % Check if all the text from the beginning
                % of the file was skiped
                if(length(str2num(nextline))~=0)
                    aux = 1;
                end;
            end
```

```
            % Load the trajectories
            %                   ind lon/lat x/y wd time
            eventData = fscanf(fid, '%d %f %f %f %f %f %lg', [7, inf]);
            eventData = eventData([2,3,6], :)';

            % Close the file
            fclose(fid);

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % We have the data for event eventIndex and starting point i %
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Clip this trajectory to the shore line %
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            index = find(eventData(:,3) <= 10);
            if(~isempty(index))
                eventData = eventData(1:index(1), :);
            end

            % Everything under the safety limit is not possible
            if(isempty(eventData))
                disp('E aberant taica!!!!');
                error('This should not be possible');
            end

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % Check for collision with the wind park %
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            index = find(inpolygon(eventData(:,1), eventData(:,2),
WindParkLimits(:,1), WindParkLimits(:,2)));

            %%%%%%%%%%%%%%%%%%%%%%%
            % Get the results %
            %%%%%%%%%%%%%%%%%%%%%%%
            if(~isempty(index)) % There is a collision
                % Say yes there was a collision
                results.outcome(eventIndex, currPt) = 1;
                % Register the time of collision in hours
                results.time(eventIndex, currPt) = (index(1)-
1)*DriftTimeInterval;
                % Register the position of colision
                results.location((eventIndex-1)*NoPoints + currPt, :) =
eventData(index(1), [1,2]);
            else
                % Say no there was no collision
                results.outcome(eventIndex, currPt) = 0;
                % Register the time of collision
                results.time(eventIndex, currPt) = Inf;
                % Register the position of colision
                results.location((eventIndex-1)*NoPoints  +  currPt,  :)  =
[NaN NaN];
            end
        end
    end

    % Go to the next event if any
    if(sum(currEvent == EndingDateTime) == length(EndingDateTime))
        lastEventProcessed = 1;
    else
        currEvent = datevec(datenum([currEvent 0 0]) + EventTimeInterval/24
+
0.00625);
        % Clean the mess
        currEvent = currEvent(1:4);
    end
end
```

```
% Plot the results
cmap1 = probabilitycol(256);
cmap2 = onecolormap(5, 230, 256, 50);

% Plot the probability to drift to WP given LoC and location of LoC
p1 = sum(results.outcome, 1)/size(results.outcome, 1);
plotcoast('Figure', '1');
plotrouteslimits('Figure', '1');
hold on
m_fill_aux(WindParkLimits(:,1),  WindParkLimits(:,2),  'r-',  'LineWidth',
1.5);
title({'{\bfProbability to drift into the wind park}', '{\bfgiven that the
boat had a LoC}', '{\bfGiven the location of LoC}'});
[x,y] = m_ll2xy(LoCPos(:,1),LoCPos(:,2));
hold on
scatter(x, y, 40, p1, 'filled');
colormap(cmap1);
colorbar;
plottowboatsinfo('Figure', '1');

figure;
bar(p1, 0.5);
hold on
axis tight
title({'{\bfProbability to drift into the wind park}', '{\bfgiven that the
boat had a LoC}', '{\bfGiven the location of LoC}'});
xlabel('LoC locations'' indices');
ylabel('Probability');
colormap(cmap1);


% Plot the probability to drift to WP given LoC and moment of LoC
p2 = sum(results.outcome, 2)/size(results.outcome, 2);
figure;
bar(p2, 0.2);
hold on
%set(gca, 'XTick', [1, 50:50:300, 313]);
set(gca, 'XTick', [1, 200:200:3130, 3130]);
axis tight
title({'{\bfProbability to drift into the wind park}', '{\bfgiven that the
boat had a LoC}', '{\bfGiven the time of LoC}'});
xlabel('LoC events');
ylabel('Probability');
colormap(cmap1);

% Plot the probability to drift to WP given LoC
p3 = sum(p2)/length(p2);
fprintf('The probability to drift into the wind park given that the boat
had a LoC: %f\n', p3);
```

APPENDIX B

This section will give a complete description of the towboats database as it was used in the simulation. The Matlab code is included here since in chapter 2 was already given a short description of the towboats.

```
% Create the towboats database as a global variable
global towboats

% Create only if necessary
if(length(towboats) == 0)
    towboats.noTowboats = 3;

    %========== NEUWERK ========%
    towboat.name = 'Neuwerk';
    towboat.length = 78.91; % in meters
    towboat.breath = 18.63; % in meters
    towboat.draught = 5.79; % in meters
    towboat.power = 5800; % in kW
    towboat.addPower = 2600; % in kW
    towboat.maxspeed = 14.0; % in knots
    towboat.minspeed = 8.50; % in knots
    towboat.speed = 11.25; % in knots
    towboat.crew = 16; % minimum crew 16 men
    towboat.tugPower = 1100; % in kN
    % ... others

    towboat.baseName = 'Cuxhaven';
    towboat.baseLocation = [53.90, 8.68]; % [lat, lon]
    towboat.location = [54.3, 8.2833]; % [lat, lon] - GKSS
    % = [54.3, 8.28] - "The Wadden Sea: Maritime
    %                   Safety and Pollution
    %                   Prevention of Shipping"
    %??? = [54.9, 8.77] Internet
    towboat.fixLocation  =  0;  %  NO  fix  position  =  suppose  there  is  a
Gaussian
    towboat.fixLocationTrigger = 37; % wind speed in knots - corresponding
to 8 Bft
    % distribution that describe its position
    towboat.locationMean = [54.1010, 8.1134]; % [latmean, lonmean]
    towboat.locationVariance = [0.0388, 0.2842]; % [latvar, lonvar];
    towboat.locationCovariance = [  0.0388, -0.0624;...
                                   -0.0624, 0.2842  ];
    % The probability that the towboat need to unload before
    % going in the rescue mission
    towboat.probNeedUnloading = 0;

    towboat.alarmTime = [0.1, 0.5]; % time in hours uniformly distributed
    % between the specified values
    towboat.reactTime = [0.1, 0.5]; % time in hours uniformly distributed
    % between the specified values
    %  towboat.connectingTime  =  [1.5,  3];  %  time  in  hours  uniformly
distributed
    towboat.connectingTime  =  [1/3,  6];  %  time  in  hours  uniformly
distributed
    % between the specified values
    towboat.fullStopTime = [0.1, 1]; % time in hours uniformly distributed
    % between the specified values
    % Add the boat to the system
    towboats.boats{1} = towboat;
    %=========================%
```

```
%=========== MELLUM =========%
towboat.name = 'Mellum';
towboat.length = 80.45; % in meters
towboat.breath = 15.10; % in meters
towboat.draught = 5.80; % in meters
towboat.power = 6620; % in kW
towboat.addPower = 1150; % in kW
towboat.maxspeed = 15.0; % in knots
towboat.minspeed = 11.0; % in knots
towboat.speed = 13.0; % in knots - "The Wadden Sea: Maritime
%             Safety and Pollution
%             Prevention of Shipping"
% 16.0 EU site
towboat.crew = 16; % minimum crew 16 men
towboat.tugPower = 1100; % in kN
% ... others

towboat.baseName = 'Wilhelmshaven';
towboat.baseLocation = [53.51, 8.12]; % [lat, lon]
towboat.location = [54.1167, 7.7833]; % [lat, lon] - GKSS
% = [54.12, 6.78] - "The Wadden Sea: Maritime
%                    Safety and Pollution
%                    Prevention of Shipping"
towboat.fixLocation  =  0;  %  NO  fix  position  =  suppose  there  is  a
Gaussian
towboat.fixLocationTrigger = 37; % wind speed in knots - corresponding
to 8 Bft
% distribution that describe its position
%%%%towboat.locationMean = [54.1167, 7.7833]; % [latmean, lonmean]
towboat.locationMean = [53.9177, 7.6134]; % [latmean, lonmean]
towboat.locationVariance = [0.0388, 0.2842]; % [latvar, lonvar];
towboat.locationCovariance = [  0.0388, -0.0624;...
                                -0.0624, 0.2842  ];
% The probability that the towboat need to unload before
% going in the rescue mission
towboat.probNeedUnloading = 0;

towboat.alarmTime = [0.1, 0.5]; % time in hours uniformly distributed
% between the specified values
towboat.reactTime = [0.1, 0.5]; % time in hours uniformly distributed
% between the specified values
%towboat.connectingTime  =  [1.5,  3];  %  time  in  hours  uniformly
distributed
towboat.connectingTime  =  [1/3,  6];  %  time  in  hours  uniformly
distributed
% between the specified values
towboat.fullStopTime = [0.1, 1]; % time in hours uniformly distributed
% between the specified values

% Add the boat to the system
towboats.boats{2} = towboat;
%============================%


%=========== OCEANIC =========%
towboat.name = 'Oceanic';
towboat.length = 80.45; % in meters
towboat.breath = 15.10; % in meters
towboat.draught = 5.80; % in meters
towboat.power = 6620; % in kW
towboat.addPower = 1150; % in kW
towboat.maxspeed = 17.5; % in knots
towboat.minspeed = 12.5; % in knots
towboat.speed = 15.0; % in knots - "The Wadden Sea: Maritime
                      %             Safety and Pollution
                      %             Prevention of Shipping"

towboat.crew = 16; % minimum crew 16 men
```

```
    towboat.tugPower = 1800; % in kN % - "The Wadden Sea: Maritime
                                     %   Safety and Pollution
                                     %   Prevention of Shipping"
    % ... others
    towboat.baseName = 'Cuxhaven';
    towboat.baseLocation = [53.90, 8.68]; % [lat, lon]
    towboat.location = [54.0667, 7.1833]; % [lat, lon] - GKSS
                        % = [54.07, 7.18] - "The Wadden Sea: Maritime
                        Safety and Pollution Prevention of Shipping"
    towboat.fixLocation = 1; % Fix location
    % If the weather is worst than
    towboat.fixLocationTrigger = 19; % wind speed in knots - corresponding
to 5 Bft
    towboat.locationMean = [0, 0]; % [latmean, lonmean];
    towboat.locationVariance = [1,1]; % [latvar, lonvar];
    towboat.locationCovariance = [  1, 0;...
                                    0, 1  ];
    % The probability that the towboat need to unload before
    % going in the rescue mission
    towboat.probNeedUnloading = 0;

    towboat.alarmTime = [0.1, 0.5]; % time in hours uniformly distributed
    % between the specified values
    towboat.reactTime = [0.1, 0.5]; % time in hours uniformly distributed
    % between the specified values
    %towboat.connectingTime = [1.5, 3]; % time in hours uniformly
distributed
    towboat.connectingTime = [1/3, 6]; % time in hours uniformly
distributed
    % between the specified values
    towboat.fullStopTime = [0.1, 1]; % time in hours uniformly distributed
    % between the specified values

    % Add the boat to the system
    towboats.boats{3} = towboat;
    %==============================%

    clear towboat
end
```

APPENDIX C

This section gives the list of impossible events as described in chapter 2.

Northern Accident – Northern Route

List of impossible events: 1995010608, 1995010712, 1995010816, 1995011800 1995011904, 1995012008, 1995012112, 1995012216, 1995012320, 1995012500, 1995012604, 1995020100, 1995020204, 1995020308, 1995020412, 1995020516, 1995020620, 1995021320, 1995021500, 1995021604, 1995021708, 1995021812, 1995021916, 1995022020, 1995022200, 1995022304, 1995022408, 1995022512, 1995030100, 1995030204, 1995030308, 1995030516, 1995030620, 1995030800, 1995030904, 1995031008, 1995031112, 1995031216, 1995031320, 1995031500, 1995031604, 1995031708, 1995031812, 1995040604, 1995042108, 1995042212, 1995042316, 1995050404, 1995050508, 1995050612, 1995050716, 1995050820, 1995052608, 1995052712, 1995052816, 1995052920, 1995053100, 1995060416, 1995060520, 1995061920, 1995062100, 1995070812, 1995071512, 1995071616, 1995071720, 1995071900, 1995072004, 1995072108, 1995072808, 1995090704, 1995090808, 1995090912, 1995091016, 1995091120, 1995091300, 1995091404, 1995091508, 1995091612, 1995091716, 1995091820, 1995100400, 1995100504, 1995100608, 1995100712, 1995100816, 1995100920, 1995101100, 1995101204, 1995101308, 1995101412, 1995101516, 1995101620, 1995101800, 1995102320, 1995102500, 1995102604, 1995102708, 1995102812, 1995102916, 1995103020, 1995111008, 1995111112, 1995111216, 1995111320, 1995111500, 1995111604, 1995112408, 1995112512, 1995112616, 1995112720, 1995112900, 1995113004, 1995120108, 1995120212, 1995120316, 1995120420, 1995120704, 1995120808, 1995120912, 1995121016, 1995122908, 1995123012, 1995123116.

Northern Accident – Southern Route

List of impossible events: 1995010116, 1995010220, 1995010400, 1995011100, 1995011204, 1995011308, 1995011412, 1995011516, 1995012320, 1995012500, 1995012604, 1995012708, 1995012812, 1995012916, 1995013020, 1995020516, 1995020620, 1995020800, 1995020904, 1995021008, 1995021112, 1995021708, 1995021812, 1995021916, 1995022020, 1995022200, 1995022408, 1995022512, 1995022616, 1995022720, 1995030100, 1995030412, 1995031812, 1995031916, 1995032020, 1995032200, 1995032304, 1995032408, 1995032512, 1995032616, 1995032720, 1995032900, 1995033004, 1995033108, 1995040112, 1995040216, 1995040320, 1995040500, 1995040604, 1995040708, 1995040812, 1995040916, 1995041020, 1995041200, 1995041304, 1995041408, 1995041512, 1995041616, 1995041720, 1995041900, 1995042004, 1995042108, 1995042212, 1995042316, 1995042420, 1995042600, 1995042704, 1995042808, 1995042912, 1995043016, 1995050120, 1995050300, 1995050716, 1995050820, 1995051000, 1995051104, 1995051208, 1995051312, 1995051416, 1995051520, 1995051700, 1995051804, 1995051908, 1995052012, 1995052116, 1995052220, 1995052400, 1995052504, 1995060208, 1995060312, 1995060416, 1995060520, 1995060700, 1995060804, 1995060908, 1995061012, 1995061116, 1995061220, 1995061400, 1995061504, 1995061608, 1995061712, 1995061816, 1995061920, 1995062100, 1995062204, 1995062308, 1995062412, 1995062516, 1995062620, 1995062800, 1995062904, 1995063008, 1995070112, 1995070216, 1995070320, 1995070500, 1995070604, 1995070708, 1995070812, 1995070916, 1995071020, 1995071200, 1995071304, 1995071408, 1995072108, 1995072212, 1995072316, 1995072420, 1995072600, 1995072704, 1995072808, 1995072912, 1995073016, 1995073120, 1995080200, 1995080304, 1995080408, 1995080512, 1995080616, 1995080720, 1995080900, 1995081004, 1995081108, 1995081212, 1995081316, 1995081420, 1995081600, 1995081704, 1995081808, 1995081912, 1995082016, 1995082120, 1995082300, 1995082404, 1995082508, 1995082612, 1995082716, 1995082820, 1995083000, 1995083104, 1995090108, 1995090212, 1995090316, 1995090420, 1995090600, 1995092000, 1995092104, 1995092208, 1995092312, 1995092416, 1995092520, 1995092700, 1995092804, 1995092908, 1995093012, 1995100116, 1995100220, 1995102008, 1995102112, 1995102216, 1995110100, 1995110204, 1995110308, 1995110412, 1995110516, 1995110620, 1995110800, 1995110904, 1995111008, 1995111708, 1995111812, 1995111916, 1995112020, 1995120600, 1995121300, 1995121404, 1995121508, 1995121612, 1995121716, 1995121820, 1995122000, 1995122104, 1995122208, 1995122416, 1995122520, 1995122700, 1995122804, 1995122908,

Southern Accident – Northern Route

List of impossible events: 1995010608, 1995010712, 1995010816, 1995010920, 1995011620, 1995011800, 1995011904, 1995012008, 1995012112, 1995012216, 1995012320, 1995012500, 1995012604, 1995012812, 1995012916, 1995020100, 1995020204, 1995020308, 1995020412, 1995020516, 1995020620, 1995020800, 1995021216, 1995021320, 1995021500, 1995021604, 1995021708, 1995021812, 1995021916, 1995022020, 1995022200, 1995022304, 1995022408, 1995022512, 1995030100, 1995030204, 1995030308, 1995030412, 1995030516, 1995030620, 1995030800, 1995030904, 1995031008, 1995031112, 1995031216, 1995031320, 1995031500, 1995031604, 1995031708, 1995031812, 1995031916, 1995032512, 1995040320, 1995040500, 1995040604, 1995040708, 1995042004, 1995042108, 1995042212, 1995042316, 1995042420, 1995050300, 1995050404, 1995050508, 1995050612, 1995050716, 1995050820, 1995051000, 1995052608, 1995052712, 1995052816, 1995052920, 1995053100, 1995060104, 1995060208, 1995060312, 1995060416, 1995060520, 1995060700, 1995060804, 1995061012, 1995061116, 1995061816, 1995061920, 1995062100, 1995062204, 1995070812, 1995070916, 1995071408, 1995071512, 1995071616, 1995071720, 1995071900, 1995072004, 1995072108, 1995072212, 1995072316, 1995072420, 1995072704, 1995072808, 1995072912, 1995073016, 1995081316, 1995090704, 1995090808, 1995090912, 1995091016, 1995091120, 1995091300, 1995091404, 1995091508, 1995091612, 1995091716, 1995091820, 1995092700, 1995100220, 1995100400, 1995100504, 1995100608, 1995100712, 1995100816, 1995100920, 1995101100, 1995101204, 1995101308, 1995101412, 1995101516, 1995101620, 1995101800, 1995101904, 1995102008, 1995102320, 1995102500, 1995102604, 1995102708, 1995102812, 1995102916, 1995103020, 1995110100, 1995111008, 1995111112, 1995111216, 1995111320, 1995111500, 1995111604, 1995112304, 1995112408, 1995112512, 1995112616, 1995112720, 1995112900, 1995113004, 1995120108, 1995120212, 1995120316, 1995120420, 1995120600, 1995120704, 1995120808, 1995120912, 1995121016, 1995121120, 1995121300, 1995121716, 1995121820, 1995122208, 1995122312, 1995122520, 1995122804, 1995122908, 1995123012, 1995123116

Southern Accident – Southern Route

List of impossible events: 1995010116, 1995010220, 1995010400, 1995011100, 1995011204, 1995011308, 1995011412, 1995012320, 1995012500, 1995012604, 1995012708, 1995013020, 1995020620, 1995020800, 1995020904, 1995021008, 1995021812, 1995022616, 1995022720, 1995030412, 1995031812, 1995031916, 1995032020, 1995032200, 1995032304, 1995032408, 1995032512, 1995032616, 1995032720, 1995032900, 1995033004, 1995040112, 1995040216, 1995040320, 1995040500, 1995040708, 1995040812, 1995040916, 1995041020, 1995041200, 1995041304, 1995041408, 1995041512, 1995041616, 1995041720, 1995041900, 1995042004, 1995042600, 1995042704, 1995042808, 1995042912, 1995043016, 1995050120, 1995050820, 1995051000, 1995051104, 1995051208, 1995051312, 1995051416, 1995051520, 1995051700, 1995051804, 1995051908, 1995052012, 1995052116, 1995052220, 1995060312, 1995060700, 1995060804, 1995060908, 1995061220, 1995061400, 1995061504, 1995061608, 1995061712, 1995061816, 1995062204, 1995062308, 1995062412, 1995062516, 1995062620, 1995062800, 1995062904, 1995063008, 1995070112, 1995070216, 1995070320, 1995070500, 1995070604, 1995070916, 1995071020, 1995071200, 1995072212, 1995072316, 1995072420, 1995072600, 1995073120, 1995080200, 1995080304, 1995080408, 1995080512, 1995080616, 1995080720, 1995080900, 1995081004, 1995081108, 1995081420, 1995081600, 1995081704, 1995081808, 1995081912, 1995082016, 1995082120, 1995082300, 1995082404, 1995082508, 1995082612, 1995082716, 1995082820, 1995083000, 1995083104, 1995090108, 1995090212, 1995090316, 1995090420, 1995090600, 1995092104, 1995092208, 1995092312, 1995092520, 1995092700, 1995092804, 1995092908, 1995093012, 1995102008, 1995102112, 1995110100, 1995110204, 1995110308, 1995110412, 1995110516, 1995110620, 1995110800, 1995110904, 1995111812, 1995111916, 1995121300, 1995121404, 1995121508, 1995122000, 1995122104, 1995122416, 1995122700.