



NTNU – Trondheim
Norwegian University of
Science and Technology

Online Shape Estimation of Icebergs at Sea

Thor Helge Billington

Marine Technology

Submission date: July 2015

Supervisor: Roger Skjetne, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY

MASTER THESIS

Online Shape Estimation of Icebergs at Sea

Supervisor:

Professor Roger SKJETNE

Co-advisors:

Professor Lars

IMSLAND(ITK)

PhD candidate Petter

NORGREN(IMT)

Author:

Thor BILLINGTON

Department of Marine Technology

July 2015



PROJECT DESCRIPTION SHEET

Name of the candidate: Thor Billington
Field of study: Marine control engineering
Thesis title (Norwegian): Online estimering av geometrisk form av isfjell til havs.
Thesis title (English): Online shape estimation of icebergs at sea.

Background

The motivation for performing marine operations in the Arctic is triggered by the large estimated amount of undiscovered natural resources in the area. The region above the Arctic Circle may account for as much as 20% of the world's undiscovered oil and natural gas resources¹. There is also an interest in exploring new routes through the Arctic waters, which would dramatically reduce the travelling distance for sea craft travelling from the Atlantic Ocean to the Pacific Ocean through the Northwest Passage. Ice surveillance would then be crucial for route planning and safety.

Further motivation can be summarized as:

- Accurate information of an iceberg's shape and geometry is important for forecasting future iceberg trajectories.
- The geometry and size of an iceberg are key components in calculating the local and global loads that could be generated during an iceberg impact.
- The metacentric height (GM) of the iceberg can be found by calculating the second moment of inertia of the waterplane and the volume of displacement. The geometry needs to be known to calculate the Second moment of inertia. This can help predict the stability of an iceberg before a towing operations.

The overall objective of this project is to develop methods to estimate the subsea geometry of icebergs based on measurements taken from, for instance, images by a UAV. This is done by using computer vision from the UAV. The estimated geometry can give a volume estimation of the underwater part of the iceberg, which further can be used to give decision in offshore operations affected by icebergs. 2D iceberg models will form the basis for the background data and development of the methods. 3D methods should be derived from this.

Work description

1. Perform a literature review to provide background and relevant references on:
 - a. Iceberg incident statistics and iceberg towing in relevant Arctic regions.
 - b. How icebergs are made, types of icebergs, and why they their shapes are irregular.
 - c. DEMs and their applications. Provide background information on DEMs from Canadian iceberg data.
 - d. Use of neural networks as basis for shape modeling and parameter identification.
 - e. Methods for land-based 3D modeling and underwater seabed mapping.
 - f. Use of UAVs and remote sensing in Arctic missions.
 - g. Sensors and methods for computer vision on mobile sensor platforms.Write a list with abbreviations and definitions of terms and concepts, explaining relevant concepts.
2. Explain why geometry, shape, and stability of icebergs are important for marine operations, including how the iceberg metacentric stability can be evaluated. If possible, propose a static method

¹ U.S. Department of Energy, Energy Information Administration, "Arctic Oil and Natural Gas Potential," October 2009.

giving a roll-stability indicator of an iceberg given an estimated 2D shape. Extend this with stability margins based on statistical data. Provide also a 3D extension of the method.

3. Propose a neural network model, and use this for input and output curve fitting from geometric parameters of an iceberg, based on 2D iceberg information. The goal is to predict the 2D cross-section shape of the iceberg keel from the iceberg sail. Generalize the neural network model and method to a 3D iceberg model, at least for an example case study.
4. Develop a dynamic estimation algorithm based on the neural network to online identify the main geometric parameters of a detected iceberg. Test the estimation algorithm on a simulation of a drifting iceberg and relevant measurements.
5. Discuss use the neural network estimated model to classify icebergs, with background in the classical families of icebergs. What parameters are most important to classify?

Guidelines

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

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

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to 1) the main supervisor and 2) the external examiner, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must appear in a bound volume or a binder according to the NTNU standard template. Computer code, pictures, videos, data series, and a PDF version of the report shall be included electronically.

Start date: 15 January, 2015 **Due date:** As specified by the administration.

Supervisor: Professor Roger Skjetne
Co-advisor(s): Professor Lars S. Imsland (ITK)
PhD candidate Petter Norgren (IMT)

Trondheim, 31.05.2015

Roger Skjetne
Supervisor

Abstract

Offshore operations in the Arctic present several challenges, such as the threat of icebergs colliding with offshore structures. This thesis will investigate methods of estimating the shape and size of iceberg keel geometries. Accurate information of iceberg size and geometry is important for forecasting future iceberg trajectories, and for calculating local and global loads generated by an icebergs impact on offshore structures. The iceberg shape and size is also useful for determining its stability before a towing operation to avoid rope slippage or the iceberg rolling over.

We have proposed a method on how iceberg surface data can be used in a towing operation for predicting its stability. This has been done by using surface lengths on icebergs to estimate their draft with a 95 % confidence interval based on previous surface length and keel draft data. The estimated draft has been used to scale trapezoidal shapes to represent iceberg keels, and then used to calculate the icebergs GZ curve. We discovered that the trapezoidal shapes gave poor shape estimations for stability calculations. This was because the volume distribution of a true iceberg was represented poorly, especially close to the water surface.

We utilized an artificial neural network (ANN) in order to estimate the shape of an iceberg keel based on iceberg surface data. The (ANN) was made by 2D iceberg keel geometries from surface data, because 3D iceberg data is scarce. The original 2D iceberg continuous profile was reduced into 8 vertices for both sail and keel geometry. This created an input matrix representing the sail geometry, and a target matrix representing the keel geometry used to train the ANN with 42 2D iceberg samples. The ANN gave more accurate results than the trapezoidal method.

We have also presented a case study on rebuilding reduced keel geometry data using shape factors from an ANN . The objection of the case study was to reduce iceberg data gathered by an AUV so that it can be transmitted real time to a surface vessel for processing. It was difficult to evaluate the performance of the ANN due to the low sample size, and because the shape factors used for estimation, were also used to measure the performance.

Sammendrag

Det er flere utfordringer ved å utføre marine operasjoner i de arktiske områdene. Håndtering av isfjell er en slik utfordring, ettersom de utgjør en trussel for skip og offshorestrukturer. Dersom isfjellene oppdages i tide kan de fjernes ved hjelp av slepefartøy. Med utgangspunkt i dette vil vi i denne oppgaven fokusere på måter man kan estimere størrelsen og formen på undersiden av isfjell. Dersom man vet hvordan undersiden av et isfjell ser ut, kan man ved hjelp av en dynamisk modell predikere hvordan isfjellet vil drive. Informasjon om undersiden av isfjellet kan også brukes til å regne på globale og lokale laster skapt fra sammenstøt mellom isfjell og offshorestrukturer. En tredje praktisk bruk av isfjellets geometri er ved slepeoperasjoner, der isfjellets form kan brukes til å beregne isfjellets stabilitet. Slik kan man unngå at isfjellet velter, eller at slepetauet mister taket.

Vi har foreslått en metode der isfjellets overflate kan brukes for å predikere isfjellets stabilitet. Dette har blitt gjort ved å bruke isfjellets diagonale overflatelengde for å estimere dypgangen til isfjellet innenfor et 95 % konfidensintervall. Dette intervallet er basert på diagonale overflatelengder med tilhørende dypganger fra et spesifikt arktisk område. Den estimerte dypgangen har så blitt brukt til å skalere en trapezoideform for å representere isfjellets underside. Vi har deretter brukt denne trapezoideformen til å regne ut isfjellets GZ-kurve for forskjellige rullvinkler. Vi oppdaget at trapezoideformen ga dårlige estimater for stabilitetsbergeningene, ettersom den representerte fordelingen av volumet dårlig, spesielt nær overflaten.

Et kunstig nevralt nettverk har blitt laget for å estimere 2D-profiler av undersiden til isfjell utifra overflatedata. Vi har brukt 2D-data ettersom 3D-data ikke var tilgjengelig. De originale, kontinuerlige 2D-profilene til isfjellene har blitt redusert til kun 8 punkter for overflaten og 8 punkter for undersiden av isfjellet. De 16 punktene ble brukt til å lage en inputmatrise og en targetmatrise for å trene det

nevrale nettverket. Det nevrale nettverket ga bedre resultater enn trapezoideformmetoden.

Vi har også presentert en måte å gjenoppbygge en redusert isfjellgeometri ved bruk av formfaktorer med et nevralt nettverk. Motivasjonen var å forenkle geometrien innhentet fra AUVen, slik at den kan sendes sanntid via den hydroakustiske linken til overflatefartøyet. Det var vanskelig å evaluere ytelsen til nettverket ettersom datasettet vi brukte var svært begrenset, og ettersom vi brukte de samme formfaktorene for å estimere som for å måle ytelsen til nettverket.

Contents

Abstract	iv
Sammendrag	vi
Abbreviations	xi
1 Introduction	1
1.1 New developments	2
1.2 Contributions	3
1.3 The online aspect	4
1.4 Outline of the thesis	5
2 Background	9
2.1 Icebergs	9
2.2 Iceberg incident statistics and towing	13
2.3 Digital models of icebergs	19
2.4 Sensors and methods for computer vision on mobile sensor platforms.	20
2.5 Unmanned aerial vehicles as a sensor platform	26
3 Application of Iceberg Keel Geometry Estimation	29
3.1 Iceberg shape geometry and trajectory calculations	29
3.2 Iceberg shape geometry and global loads	31
3.3 Iceberg shape and stability	31
3.4 An example of how iceberg shape data has been used for stability estimation	34
3.5 Discussion	40
4 Theoretical Background	45
4.1 Neuron model	46
4.2 Transfer functions	48
4.3 Network architecture	49
4.4 Training the network	49
4.5 Generalization	54

5	Case study: Estimation of Keel Geometry from Surface Data	59
5.1	Data collection and preprocessing	60
5.2	Selecting the architecture	62
5.3	Training the network	63
5.4	Geometric shape results	72
5.5	Discussion and improvements	78
6	Case study: Online Estimation of Keel Geometry from an AUV Communication Link	87
6.1	Preprocessing	88
6.2	The network	88
6.3	Discussion and improvements	96
6.4	Conclusion	96
A		101
Bibliography		103

Abbreviations

UAV	U n m anned A erial V ehicles
DEM	D igital E levation M odel
IM	I ce M anagement
GPS	G lobal P ositioning S ystem
INS	I nertial N avigation S ystem
UUV	U n m anned U nderwater V ehicle
CV	C omputer V ision
SIIS	S ub s urface I ce I ntelligence S ystems
GIS	G eographic I nformation S ystem
DRC	D eliberative R eactive C ontrol system
IPC	I mage P rocessing C ontrol system
PFC	P rimary F light C ontrol system
GSD	G round S ample D istance
SAR	S ynthetic A perature R adar
RADAR	R adio D etection and R anging
SONAR	S ound N avigation and R anging
IM	I ce M anagement
HCSM	H ierarchial C oncurrent S tate M achine

Chapter 1

Introduction

The motivation for carrying out marine operations in the Arctic is triggered by estimations of large amounts of undiscovered natural resources in the area. The region above the Arctic Circle may account for as much as 20 % of the worlds undiscovered oil and natural gas resources. Another motivation for carrying out marine operations in the Arctic is exploring new shipping routes, which could dramatically reduce the travelling distance and transit time for ship traffic. Ice surveillance would then be crucial for route planning and safety.

When performing offshore operations in the Arctic, there are several challenges. One of those challenges is the threat of icebergs on offshore structures and vessels. Icebergs can exert extremely high loads on vessels, offshore platforms, and seabed installations [Timco, 1999]. If an iceberg is detected, there are several ways to relocate it, so it no longer poses a threat to the offshore structures. The most common practice is towing. We can summarize the motivation for investigating iceberg size and shape as:

- Accurate information of iceberg size and geometry is important for forecasting future iceberg trajectories.
- The geometry and size of icebergs is also key components in calculating the local and global loads that could be generated during an iceberg impact.
- The iceberg's distribution of volume (shape) determines its stability. An iceberg keel shape estimation can therefore help predict the stability of an iceberg before a towing operation.

The overall objective of this thesis is to estimate the geometric shape of the iceberg keel from surface data. The shape estimation method will be a part of an ice management system within ice surveillance. In Figure 1.2 we can see a proposed schematic of an ice management operation. If we take Figure 1.2 as a basis for management of icebergs, this research falls under the detection, tracking and classification block.

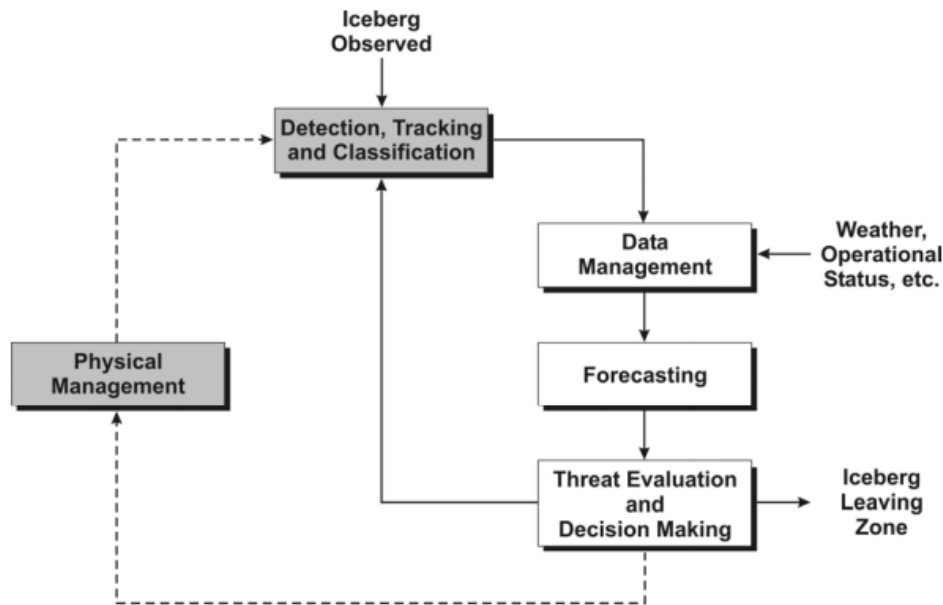


FIGURE 1.1: Ice management schematic given by C-CORE [2007].

1.1 New developments

Today an underwater sensor platform is needed to gather information on iceberg keels. Scientists have used underwater vehicles to scan icebergs using sonar. This is time-consuming and requires a lot of resources.

We will propose an ice surveillance system where one first creates a large enough data set of iceberg sail geometries and corresponding keel geometries to create a statistical foundation for estimation. If estimation from the data set gives sufficient estimations of the iceberg keel, the underwater sensor platform does not need to be present all the time. The underwater platform can then just be present for a limited period to build up a sufficient data set. The underwater sensor system can instead work as a backup solution if iceberg keel estimations are not sufficient.

The goal is to create a cloud-based neural network where industry stakeholders and research institutions can share their iceberg data to a common server. This data will be used as a statistical basis for estimation tools, such as a neural network for estimating iceberg keel geometries. The network will increase its performance the more data samples it has available. The community cloud may be organized by one of the organizations itself or by incorporation of a third party.

During the last couple of decades, small unmanned aerial vehicles have become an inexpensive platform to carry electro-optical (EO) and infrared (IR) cameras. These unmanned aerial vehicles can be used to perform ice surveillance and create digital elevation models of icebergs from photogrammetry.

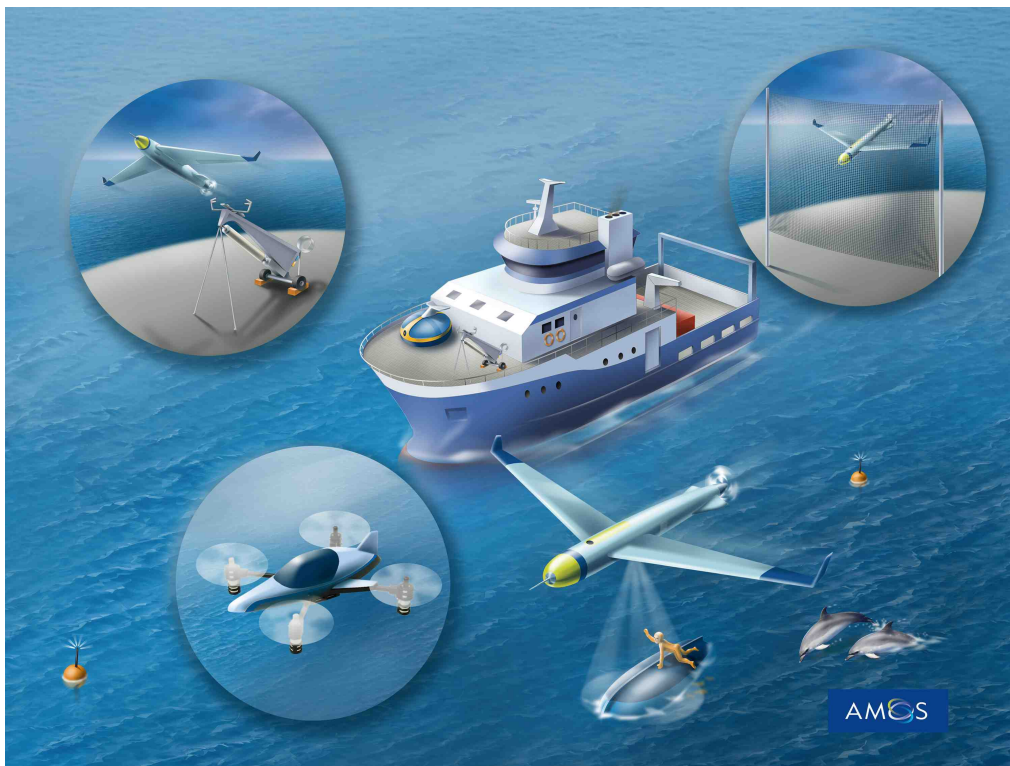


FIGURE 1.2: Picture taken from the centre for Autonomous Marine Operations and Systems (AMOS) illustrating aerial vehicles and a surface vessel.

1.2 Contributions

We have investigated how iceberg stability can be estimated by iceberg surface lengths. This was done in 2D, but we have included a section on how the method also can be applied to a 3D iceberg keel.

Later on we will propose a neural network for function approximation that can estimate iceberg keel shapes from their corresponding sail shapes. The performance of the network is dependent on the data size. Today the Program for Energy Research and Development (PERD) database has gathered a comprehensive set of iceberg data, but for this project to be viable, more data needs to be gathered, especially on 3D iceberg sails and keels.

Finally we have investigated how the high resolution iceberg keel geometry data can be simplified before being transmitted by an autonomous underwater vehicle (AUV) to a surface vessel. The acoustic link between a surface vessel and AUV is limited for a real-time marine operation. We will see how well the iceberg keel geometry can be rebuilt by a neural network from only a few shape factors. Perhaps the AUV will only need to transmit a couple of shape factors during an operation instead of a high resolution geometry.

1.3 The online aspect

The motivation for this thesis is to create an estimation tool for iceberg keels to use during marine operations. This means operating with a time constraint for the estimation method. However, for a neural network the estimation itself does not take a lot of computational time. It is the training algorithm of the neural network that takes time. Therefore the training of the network and gathering of iceberg data must happen offline before using it in marine operations. The online time constraint for a neural network will be the communication link between the mobile sensor platform and the computer onboard the surface vessel. Even if the training algorithm used less computational time for an online operation, it would not have any new keel data for the network to use.

However, if an AUV has gathered new keel data from icebergs with corresponding sail data then the data can be added to the statistical basis. The network can then be retrained offline with the new data before being used in a marine operation.

1.4 Outline of the thesis

Chapter 2 gives background information on iceberg statistics, towing and use of different sensors in the Arctic.

Chapter 3 describes some application of the use of iceberg keel geometry, and gives an example of how it can be used in a towing operation.

Chapter 4 presents some theoretical background on artificial neural networks.

Chapter 5 presents a case study of estimating iceberg keel geometry from surface data using an artificial neural network. We also discuss the results from the study.

Chapter 6 presents a case study on rebuilding reduced keel geometry data using shape factors with a neural network. The motivation is to simplify the data gathered by an AUV so that it can be transmitted real time to a surface vessel for processing.

Chapter 7 gives a final conclusion of the work carried out in the thesis and gives recommendations for further work.

Definitions

Digital elevation model

A digital elevation model (DEM) is a digital model or 3D representation of a terrain's surface. A DEM can be represented as a raster (a grid of squares, also known as a heightmap when representing elevation) or as a vector-based triangular irregular network (TIN). DEMs are commonly built using data collected using remote sensing techniques, but they may also be built from land surveying. DEMs are often used in geographic information systems and are the most common basis for digitally-produced relief maps.

First-year ice

Sea water freezes to create sea ice. All sea ice that has not survived one summer's melt is called first-year ice.

Photogrammetry

Photogrammetry is the science of making measurements from photographs, especially for recovering the exact positions of surface points. Moreover, it may be used to recover the motion pathways of designated reference points located on any moving object, on its components and in the immediately adjacent environment.

Spatial statistics

spatial statistics includes any of the formal techniques that study entities using their topological, geometric, or geographic properties.

Geographic information system

A geographic information system (GIS) is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of spatial or geographical data.

Computer vision

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images and high-dimensional data from the real world.

Iceberg detection and monitoring

Detection of icebergs in a relevant proximity to the protected operation, vessel or installation. The detected icebergs must be monitored continuously to evaluate the threat. Detection and monitoring are performed by visual observation, marine radars, aircraft radars and satellite (radar images). Radars also work in bad weather and are essential for continuous monitoring.

Iceberg management system

The complete system with the aim to mitigate the risk of icebergs is called an iceberg management system. The system consists of iceberg detection, monitoring and evaluation of potential threats. If the iceberg is considered a threat, physical iceberg management, called iceberg handling is needed.

Ice intelligence

An ice intelligence system is one of several required elements in an IM system. The ice intelligence system has to ensure that all information regarding ice conditions that might influence marine operations is collected and presented for relevant personnel in due time. For ice intelligence, typically surface scouting tools such as satellites, airborne recognizance, marine radars, drift buoys and visual observations from icebreakers are used.

Ground sample distance

In remote sensing, ground sample distance (GSD) in a digital photo (such as an orthophoto) of the ground from air or space is the distance between pixel centers measured on the ground. For example, in an image with a one-meter GSD, adjacent pixels image locations are 1 meter apart on the ground.[1] GSD is a measure of one limitation to image resolution, that is, the limitation due to sampling.

3D reconstruction

In computer vision and computer graphics, 3D reconstruction is the process of capturing the shape and appearance of real objects. This process can be accomplished either by active or passive methods. If the model is allowed to change its shape in time, this is referred to as non-rigid or spatiotemporal reconstruction.

LIDAR

LIDAR, which stands for Light Detection and Ranging, is a remote sensing method that uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth. These light pulses combined with other data recorded by the airborne system generate precise, 3D information about the shape of the Earth and its surface characteristics.

Visual Odometry

In computer vision, visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images.

Image processing

Image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a 2D signal and applying standard signal-processing techniques to it.

Computer stereo Vision

Computer stereo vision is the extraction of 3D information from digital images, such as those obtained by a camera. By comparing the information about a scene from two vantage points, 3D information can be extracted by examination of the relative positions of objects in the two panels.

Chapter 2

Background

This chapter provides background information on icebergs, incident statistics, iceberg classification and towing in specific Arctic regions. The chapter will also contain background information on the digital elevation models from Canadian iceberg data. We will especially focus on the use of UAVs in the Arctic, and describe sensors used for the mobile sensor platform.

2.1 Icebergs

Icebergs are fresh-water ice, composed of the glacier and ice-sheet ice that originally fell as snow. When the glacier edge reaches the sea, large pieces break off, drift away and become icebergs [[Palmer and Croasdale, 2013](#)].

The lifespan of an iceberg can be divided into different stages [[Wadhams, 2000](#)]. The Arctic icebergs first calve from the narrow, steep mountain glaciers that drain the Greenland ice sheet, other mountainous islands and the coasts of the Arctic. These icebergs are usually smaller and more randomly shaped than Antarctic icebergs, as the glaciers are narrow and fast flowing with crevasses [[Wadhams, 2000](#)]. The icebergs often get trapped in the fjords where their draft exceeds the depth of the fjord. In the fjord, the temperature differences between depths may change the profile of the iceberg. This results in the icebergs losing stability and capsizing in their early lifespan [[Wadhams, 2000](#)].

Icebergs in the Grand Banks area of Canada are weathered and frequently tilted and rolled. It has been implied by [Timco \[1999\]](#) that the surface length of the

iceberg surface area can be used to determine the profile of the iceberg keel. The stability of the iceberg must ensure a certain distribution of iceberg volume which creates a relation between keel draft and length in the cross section.

Iceberg shape and size

A popular characterization of iceberg shapes is given in Figure 2.1. Although the iceberg classification in Figure 2.1 is simple, it is not very useful for engineering purposes as we will discuss in the next section. McKenna [2004] presents more complex models which include keel shape, and sail to keel volume ratios.

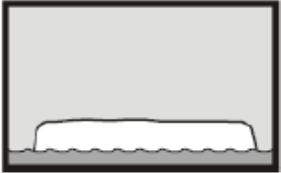


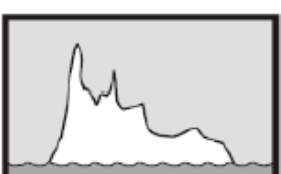
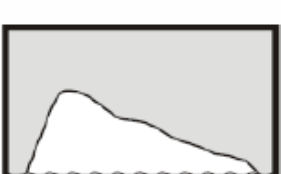

	<p>Tabular iceberg Iceberg with a flat top, steep vertical sides and a length-to-height ratio above 5:1. They look like rectangular prisms.</p>
	<p>Blocky iceberg Iceberg with a flat top and steep vertical sides, similar to a giant ice block. Length-to-height ratio between 3:1 and 5:1</p>
	<p>Dome iceberg Iceberg with rounded top, resembling a dome.</p>
	<p>Pinnacle iceberg Iceberg with one or more spires or peaks.</p>
	<p>Wedged iceberg Iceberg with one steep vertical side, resembling a giant wedge.</p>
	<p>Dry-dock iceberg Iceberg with an eroded center, with two or more separate sides above the water.</p>

FIGURE 2.1: Different iceberg shape classifications.

Icebergs can be extremely large. Excluding ice islands, the largest recorded iceberg was observed off the Baffin Islands in 1882. The iceberg was 13 km \times 6 km in size with a freeboard of 20 meters [Wadhams, 2000]. However, most icebergs will be in the range of 100-300 meters in diameter. Table 2.1 presents the size classification of icebergs by mass. In Figure 2.2 we see iceberg size compared to offshore structures and vessels.

TABLE 2.1: Table of iceberg classification based on mass height and length.

Type	Mass [T]	Sail height [m]	Sail length [m]
Growler	500	\langle 1	\langle 5
Bergy bit	1,400	1-5	5-15
Small berg	100,000	5-15	15-50
Medium berg	750,000	15-50	50-100
Large berg	5,000,000	50-100	100-200
Very large berg	\rangle 5,000,00	\rangle 100	\rangle 200

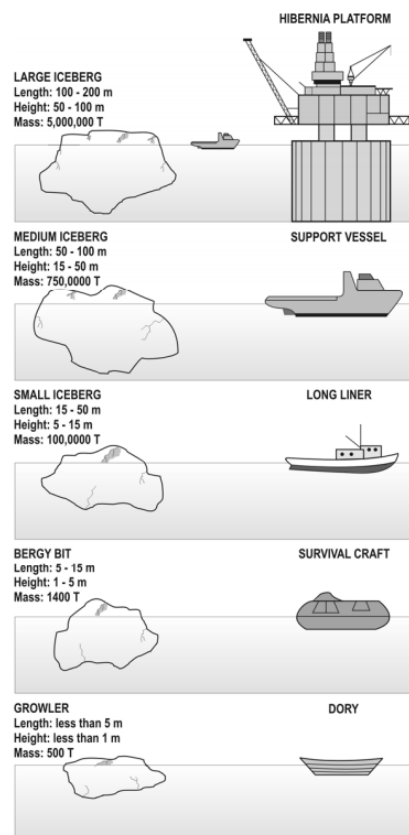


FIGURE 2.2: A general classification of iceberg sizes [C-CORE, 2007].

Discussion on iceberg classification

The iceberg classifications that are given by [Keys \[1986\]](#) and [Figure 2.1](#) do not classify iceberg shape with parameters one can measure. This is because named classes do not contain a mathematical shape definition. One can argue that classification done by names are only useful if physical measurements are not available.

When only relying on eye-sightings, the traditional classification method is good, and will give us a rough idea of how the shape is and will be useful in a colloquial manner. However, for engineering purposes, the shape description given in [Figure 2.1](#) is not useful for trajectory calculations, stability calculations or calculations on local load impacts from an iceberg. The classes are not divided and defined by measurable geometric parameters. For instance, in [Figure 2.1](#) we see the classification "Dome iceberg", which is described as an iceberg with a rounded top. The question is then how to quantify this roundness and compare it to other classes. One can ask, when does the shape change from a Dome shape to a tabular shape iceberg? Since these definitions have not been established, it is difficult to justify the use of creating a new classification model connected to physical parameters.

2.2 Iceberg incident statistics and towing

The presence of icebergs makes the extraction and exploration of oil and gas in the Grand Banks area difficult. The threat caused by icebergs affects production platforms, collection and offloading systems and exploration schedules [[Timco, 2007](#)]. Every year there is a risk involving icebergs that threaten marine operations. The icebergs range from large icebergs with a mass of one million tons to small growlers (see [Figure 2.1](#) for definition). The strong wind and wave conditions together with poor visibility can cause iceberg monitoring and handling of the icebergs to be difficult. The main monitoring resources used are radar, fixed-wing aircrafts, and remote sensing satellites. For iceberg handling, support vessels are mostly employed for towing, although both explosives and water cannons have been used to deal with icebergs [[Timco, 2007](#)].

The Program for Energy Research and Development (PERD) database of icebergs contains 1750 records of 46 individual fields on iceberg management operations conducted between 1973 and 2012 from the Grand Banks area and Labrador. The

Iceberg operational data was separated from Labrador and Grand banks because of the significantly different approaches to ice management in the two areas. This is primarily related to the much larger ice zones associated with anchored rigs on the Grand Banks.

In Figure 2.3 the number of icebergs crossing latitude 48°N (about 30 km north of St. Johns) has been monitored by the U.S. Coast Guard International Ice Patrol.

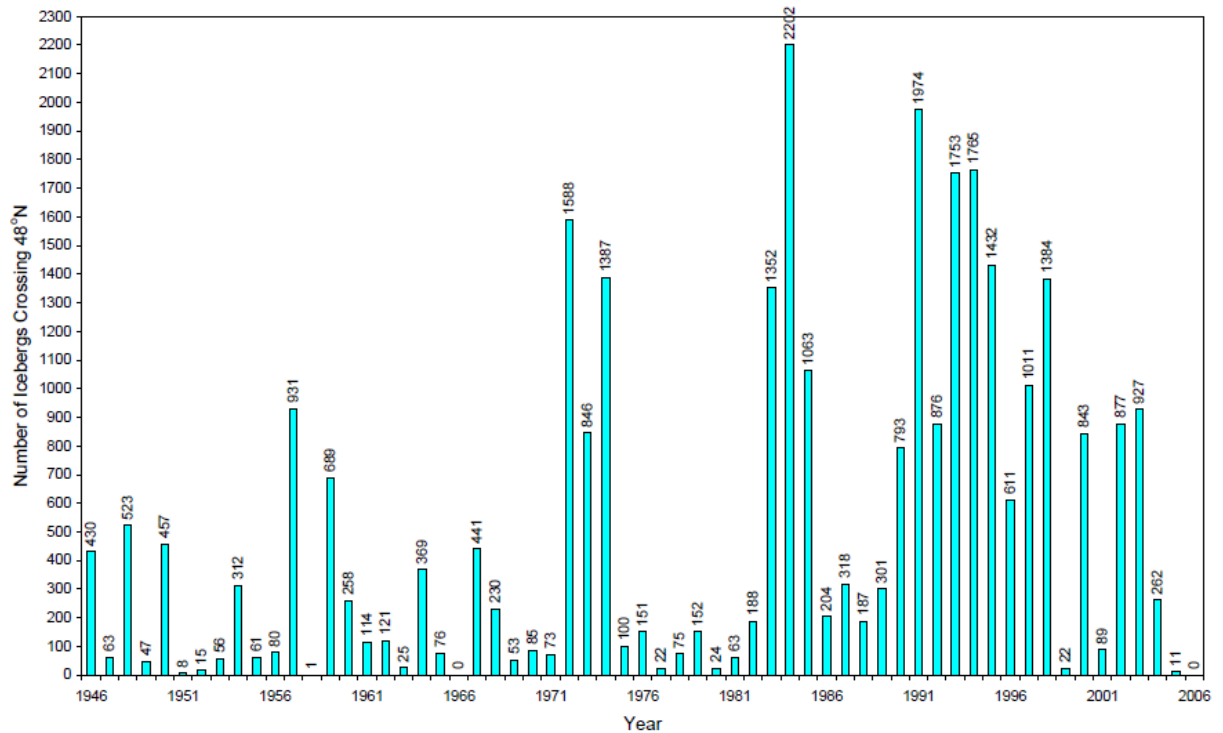


FIGURE 2.3: Annual number of iceberg crossing 48°N gathered by the U.S. Coast Guard International Ice Patrol.

There is a large intra-annual variability in iceberg occurrence. Figure 2.4 shows the mean and median crossings in the period from 1946 to 2006. The greatest averages occur in the months of April (130 crossings) and May (143 crossings), while in September, October, November, and December there are almost no iceberg sightings.

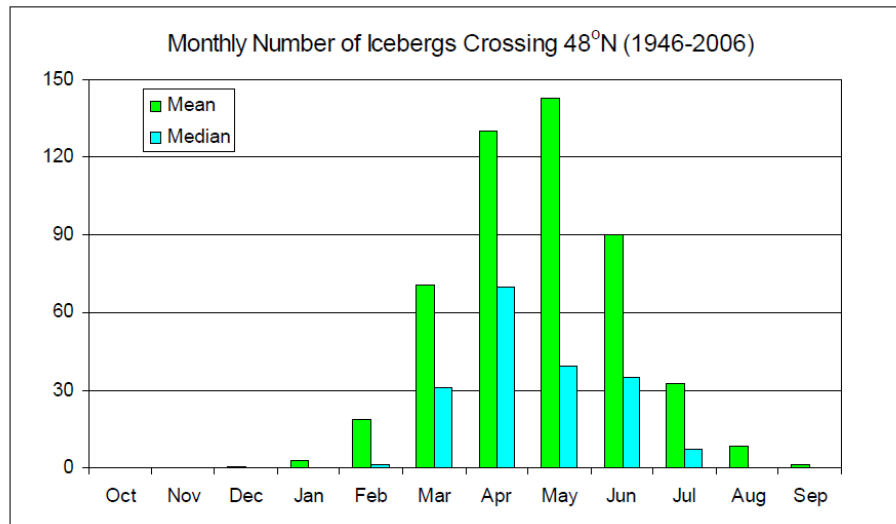


FIGURE 2.4: Monthly number of iceberg crossing 48 ° N gathered by the U.S. Coast Guard International Ice Patrol.

Iceberg towing

The PERD Comprehensive Ice Management Database (CIMD) was established with the goal of gathering all logged iceberg data from Grand Banks and Labrador, since the early 1970's, into one centralized library. In this section, we present data from this database, which is most relevant for iceberg towing.

In Figure 6.1 we can see what the ending reason, meaning how the iceberg operation was stopped, for 1750 recorded iceberg management operations.



FIGURE 2.5: Data taken from the PERD Comprehensive Management Database

Most iceberg management operations ended by tow release (58 %) which means the operations ended because it achieved a desired result. The second most common reason was rope slippage (23 %) and the third most common reason was rolling of the iceberg (9 %). The report does not state whether or not a new attempt was tried in unsuccessful operations.

Effect of tow force

Table 2.2 presents 873 records of tow force from the PERD CIMD. The report by Timco [2013] claims that the success of iceberg operations is more dependent on iceberg size and stability than tow force applied during operation. The probability of operational success increases with the strength of the tow force, although there is evidently an anomaly for forces above 100 tons. This can be explained by the small number of occurrences for forces above 100 tons.

TABLE 2.2: Tow force here is given in tons.

Force	Number of Records	Percentage of Records (%)	Mean Time	Number Successful	Percent Successful (%)
<10	38	4.4	5.9	26	68.4
10-25	201	23.0	7.7	122	60.7
26-50	357	40.9	11.4	279	78.2
51-75	185	21.2	15.8	148	80.0
76-100	77	8.8	18.1	68	88.3
>100	15	1.7	23.4	7	46.7

Tow success versus shape

In Table 2.3 the database shows success rates of 1750 iceberg operations when comparing different iceberg shapes. The table defines two types of success; technical tow success and calculated tow success. Technical tow success is defined as whether or not the iceberg changes course so it no longer causes a threat. Calculated tow success is defined by a number of criteria to each of which are assigned separate values. The sum of this score rates the success of the operation [Timco, 2013]. The key fields are:

- Number of connection attempts.
- Vessel tow heading versus the iceberg course corrected.
- If there was a change in the iceberg drift course.
- If there was a change in the iceberg drift speed.
- The outcome of the operation.

The lowest calculated success rate is for wedged shaped icebergs with a 66.3 % success rate. The highest success rate is for non-tabular shaped iceberg with 80 % success rate, but this might just be because its low samples size of 10 iceberg tows. Timco [2013] does not state why the success rates vary between different shapes. In general, the success rate does not vary a lot due to shape, where most technical success rates are between 80 and 90 %.

TABLE 2.3: Table of iceberg shape vs. tow success for Grand Banks and Labrador.

Shape	Total # of Records	Percentage of Total (%)	# of Calculated Successful Tows	Percentage of Calculated Successful Tows	# of Technically Successful Tows (%)	Percentage of Technically Successful Tows (%)
Wedge	101	5.8	67	66.3	83	82.2
Tabular	218	12.5	167	76.6	196	89.9
Non-Tabular	10	0.6	8	80.0	10	100
Dome	291	16.6	203	69.8	251	86.3
Pinnacle	437	25.0	293	67.0	401	91.8
Dry-Dock	449	25.7	345	76.8	403	89.8
Blocky	85	4.9	65	76.5	77	90.6
Unknown	159	9.1	117	73.6	128	80.5

Tow success versus size

In Table 2.4 we see the success rate of iceberg towing versus size. One of the largest success rates are for very large icebergs, but this might be due to the low sample size for very large icebergs. From large icebergs to growlers, we see a trend that smaller icebergs have less success rate. This might be because the smaller icebergs tilt easier causing tow slippage or rolling of the iceberg. However, the report by [Timco \[2013\]](#) does not state a reason for the different success rates.

Shape	Total # of Records	Percentage of Total (%)	# of Calculated Successful Tows	Percentage of Calculated Successful Tows (%)	# of Technically Successful Tows	Percentage of Technically Successful Tows (%)
Very Large	8	0.5	6	75.0	6	75.0
Large	337	19.3	250	74.2	308	91.4
Medium	525	30.0	383	73.0	482	91.8
Small	627	35.8	436	69.5	553	88.2
Bergy Bit	117	6.7	88	75.2	97	82.9
Growler	43	2.5	32	74.4	35	81.4
Unknown	93	5.3	70	75.3	68	73.1

TABLE 2.4: Here we compare iceberg size versus tow success.

2.3 Digital models of icebergs

The data is taken from the NRC-PERD Iceberg Sightings Database. The database is a result of a major effort to collect visual and radar-detected iceberg sightings from ships, offshore structures and satellites for the past 400 years. The database is a compilation of iceberg sighting entries from the years 1619 to 2012 and has become the industry standard for historical iceberg sightings in Newfoundland and Labrador.

The database includes the date and time of sightings with geographic coordinates. It also provides iceberg shape and size categories with dimensions if available along with information about the source of the sighting.

Overview of database

- 3D geometry for 28 iceberg keels from sonar profiling.
- 3D geometry for 566 sails, from stereo photography (digital or image file).
- 2D profiles for 155 iceberg sails and keels (image files).
- 3D animations for full views of 76 icebergs.

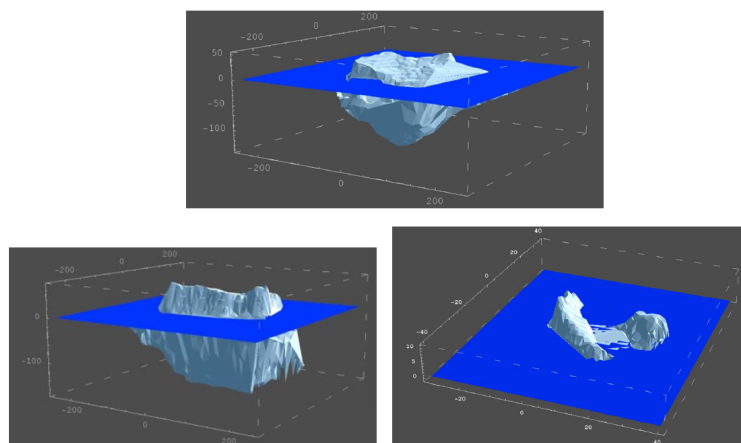


FIGURE 2.6: Examples of 3D geometries from the iceberg shape database [Timco \[1999\]](#) .

Types of data in the PERD database

Here we will summarize the types of data which is available in the PERD database [Timco, 1999].

Type 1: 3D data on iceberg keel and sails.

- This data has been gathered through three studies using sonar sidescan data. The studies are: 1984 and 1985 Dobrocky sidescan sonar data, and Dynamics Iceberg Grounding and Scouring (DIGS) project.

Type 2: Detailed 3D data on iceberg sails.

- Five projects have used stereo-photographs to obtain a total of 563 iceberg topside contours.

Type 3: Cross sections of iceberg sails and keels.

- This is data gathered by photography and sightings, generating vertical profiles of the iceberg from two or four sides. The information was collected during iceberg towing and is the data used in this thesis for verification of the developed theory.

Type 4: Sketches of iceberg sails.

- There are three reports providing data on iceberg sails from hand drawn contours.

2.4 Sensors and methods for computer vision on mobile sensor platforms.

3D modeling is the process of developing a mathematical representation of any 3D surface of an object. In this section we will discuss the use of computer vision for creating a 3D model of an iceberg.

The use of computer vision involves several processes. Light falling on a scene, is reflected, gathered by a lens and then turned into a digital image processed by different algorithms to extract the required information. For the purpose of ice surveillance, we want to use computer vision to create digital elevation models (DEMs) of icebergs [Corke, 2011].

McKenna [2004] states that iceberg shape data comes essentially from three sources:

- Ship based photographs.
- Aerial photogrammetry.
- Underwater sonar profiles.

The photographs provide accurate 2D profiles while photogrammetry can give accurate 3D geometries for the proportion of the iceberg that extends above the waterline. The underwater sonar can provide either 2D or 3D geometries. The 3D geometries are often created by point clouds by identifying the same pixel in several photo frames, and then finding the distance to each pixel by triangulation.

Today iceberg surveillance relies on visual confirmation when weather conditions permit. However, the regions infested with ice and icebergs are often characterized by considerable fog, cloud, and precipitation. This renders visual detection useless, leaving radar-based systems as the primary surveillance tool [C-CORE, 2007].

There are many radar-based surveillance systems that can be used for IM. Some examples are microwave radar mounted on the support vessels, satellite radar, shore-based high-frequency radar, and aircraft-based systems. Figure 2.7 contains an overview of the available sensors for iceberg detection.

SENSOR	PLATFORM				
	Ice Management or Supply Vessel	Production or Drilling Facility	Fixed Wing Aircraft	Helicopter	Satellite
Visual	X	X	X	X	
Terrestrial Microwave Radar	X	X			
Synthetic Aperture Radar			X		X
Search Radar (in its various forms)			X		
High Frequency Radar [†]		X			
Thermal Infra Red	X	X	X		
Optical (Camera)	X	X	X		X
Sonar	X	X			

FIGURE 2.7: Overview of the available sensors for iceberg detection for a platform [C-CORE, 2007].

In Figure 2.8 we see a comparison of the different attributes in terms of cost, coverage, spatial, and temporal resolution [Haugen, 2014].

Platform	Coverage	Resolution		
		Space	Time	Cost per area
Satellites	Excellent	Intermediate	Coarse	Intermediate
UAVs	Very good	Fine	Intermediate	Intermediate
Shipboard	Low	Intermediate	Fine	Coarse
Buoys	High	Sparse	Intermediate	Fine
USVs	Intermediate	Sparse	Low	Fine
Sub-sea	Good	Very fine	Intermediate	

FIGURE 2.8: A sensor platform parameter comparison by [Haugen \[2014\]](#).

Ice monitoring system

As mentioned, when monitoring icebergs in the Grand Banks there must be an ice intelligence system in place to detect and track the icebergs close to the offshore installations with sufficient confidence. There are several sensor platforms to use for ice monitoring, each with different weaknesses and strengths. Sensors vary in resolution, range, update frequency, and accuracy. With the resolution of remote sensing, one often refers to the ground sample distance (GSD) which describes the distance between two sampled pixels. It is important to have an adequate GSD to be able to distinguish the object from the surroundings.

Sensor types

In this section, we present different sensor types available for gathering information about ice and icebergs. We will comment on the strengths and weaknesses of the different devices [[Haugen, 2014](#)].

Laser

These devices emit laser pulses and measure the time of the round trips, this will then give an estimate of the distance travelled. These devices can be used for providing a topograph of the ice cover and icebergs. However, the laser is vulnerable to clouds.

Thermal infrared (TIR)

These sensors measure emissions based on temperature and can help measure ice surface temperature. This property can assist in determining ice motion, iceberg detection, and flow size distribution. Since these sensors cannot penetrate cloud covers, they can only be used as a complementary sensor in ice intelligence applications.

Active microwave

These devices emit electromagnetic signals toward the regions of interest and measure the reflections from the device again. There are three classes of frequently used active microwave sensors in remote aerial sensing. Common for all three methods is that they can penetrate clouds, polar darkness, and precipitation. The three common active microwave classes are:

- Synthetic aperture radars (SARs) have a high spatial resolution that can provide complex pictures of a region. This sensor class can distinguish between first-year and multi-year ice, and can differentiate water from ice. The SAR sensors can be used in ice classification, determining ice motion and ice deformation. However, interpretation of the SAR images can be difficult depending on the environmental conditions.
- Scatterometer measures the energy backscatter from the target surface. This can give sea ice data about ice roughness and dielectric properties of the surface. The sensor's weakness is its spatial resolution, which can be several kilometers for scatterometers on satellites.
- Radar altimeters measure altitude above the terrain presently beneath an aircraft by timing how long it takes a beam of a radio wave to reflect from the ground and return to the plane. This method can be used to evaluate the ice thickness and iceberg height by assuming hydrostatic equilibrium and

an estimation of the ice density. The limitation of this device is its area of measurement.

Acoustic sensors

The sensors are usually used underwater, with upwards looking sonar. They can create an underwater topography of the sea ice. It can also measure sea ice, and iceberg drift and distinguish between first-year and multi-year ice. The disadvantage is limited coverage and propagation delays that vary with salinity and temperature.

Meteorological suite

These are sensors that can measure temperature, wind velocity and humidity of the atmosphere. This is data that can improve other estimates of sea ice, and icebergs.

Oceanographic sensor suite

These are sensors that measure ocean temperature, salinity, and currents. This is used to determine drift and heat flux of the ice and icebergs.

Overview of different sensor platforms

The ice intelligence design relies on different sensors working together for monitoring the iceberg. Table 2.9 presents an overview of the various sensor platforms together with the different sensor types [Haugen, 2014].

Sensor type	Platform				
	Satellite	UAV	Shipboard	Buoy	Underwater
Optical (VNIR and TIR)	✓	✓	✓		
Laser altimeter/scanner	✓	✓			✓
Radiometer	✓	✓	✓		
SAR	✓	✓			
Marine radar			✓		
Scatterometer	✓	✓			
Radar altimeter	✓	✓			
Acoustic techniques			✓		✓
Meteorological suite		✓	✓	✓	
Oceanographic suite			✓		✓

FIGURE 2.9: Overview of sensor platform and sensor type by [Haugen \[2014\]](#).

2.5 Unmanned aerial vehicles as a sensor platform

One of the main reasons for the current interest in small unmanned aerial vehicles is that they offer an inexpensive platform to carry electro-optical (EO) and infrared (IR) cameras. Most small and miniature air vehicles currently deployed carry either EO or IR cameras [[Beard and McLain, 2012](#)]. UAVs also demonstrate flexibility in geographical coverage, spatial and temporal resolution [[Haugen, 2014](#)].

The UAV sensor platform can not compete with for example satellites on information cost per area. However, satellites struggle with giving accurate height estimations of icebergs and can not therefor create accurate digital elevation models. Nonetheless the UAV sensor platform has challenges regarding maintenance costs, launch, and recovery systems when dealing with the hostile environment. Such as extreme coldness, high wind speeds, and icing problems. The UAVs must also be able to operate at all times, both night and day.

A vision based sensor platform with UAVs is equipped with the following sensors for guidance navigation and control:

- Accelerometers.
- Rate gyros.
- Pressure sensors.
- Magnetometers.
- GPS.

For using a vision based sensor platform in a marine operation we are relying on a robust navigation system. Therefore we will briefly discuss GPS vulnerability in the Arctic in the next section.

GPS Vulnerability in the Arctic

A standard UAV navigation system relies on GPS and an inertial navigation system (INS). If the GPS signal falls out the state estimation solution by the INS will be unusable after a couple of seconds. Especially for small UAVs that use a low-cost INS. The UAV is therefore very vulnerable when only relying on the GPS signal for its navigation system.

One solution to the GPS vulnerability is to combine information from the INS and information from the video camera. Using visual odometry with an algorithm that registers the onboard video to a geo-referenced satellite or an aerial image has been proposed by [Conte \[2009\]](#).

[Haugen \[2014\]](#) discussed the use of a mobile sensor network for sea ice monitoring. The idea is to use a network of unmanned aerial vehicles to exploit strengths of each other as a sensor platforms. The UAVs will communicate with each other and define its path planning and placement based on the other sensor devices in the network.

Other assumptions and limitations

We have a challenge if the UAV images are taken at a flight altitude much larger than the iceberg. It will then be difficult to distinguish the iceberg height compared to the ground.

When designing UAVs for Arctic operations, it is important to map the UAVs capabilities in terms of maximum payload, range, weight, altitude, endurance, range, and speed. These parameters determine which types of sensors the UAV can carry, the length and range of each mission etc. In Norway, research is being conducted at NORUT on the use of UAVs in the Arctic environment. Based on the properties mentioned, there are three main classes of UAVs today. They are:

1. **Aerostats** is not very flexible in terms of range. However, its operational costs and launch and recovery capabilities are good. See Figure (2.10).
2. **Fixed-wing vehicles** has complicated launch and recovery capabilities but has a larger operational range.
3. **Rotary-wing vehicles** can land and take-off vertically, but has limited operational range.



FIGURE 2.10: Picture of the OceanEye system by Maritime Robotics in Trondheim, this is tethered aerostat system that offers a surveillance solution for demanding maritime environment.

Chapter 3

Application of Iceberg Keel Geometry Estimation

In this chapter we will briefly describe how iceberg shape geometry can be used for calculating iceberg trajectories, iceberg local and global load impacts, and stability. In the last section there will be given an example of how this shape data can be used for estimating the stability of an iceberg.

3.1 Iceberg shape geometry and trajectory calculations

We can estimate an icebergs trajectory by creating a dynamic drift model. The dynamic model is based on the momentum balance from Newton's second law.

The momentum balance can be given as:

$$Mass \times Acceleration = Air \ Stress + Water \ Stress + Coriolis \ Force \quad (3.1)$$

The dynamic model of iceberg drift is taken from [Smith et al. \[1987\]](#). We assume the iceberg will drift under the influence of the vector sum of air drag F_a , water drag F_w , pressure gradient force F_p in the water, Coriolis force and possibly towing force F_T . With these forces the equation of motion in the body fixed frame is:

$$M(a + f \times V) = F_a + F_w + F_p + F_T, \quad (3.2)$$

where M represents both mass and added mass of the iceberg. The shape of the iceberg affects both the air stress and water stress on the iceberg.

The iceberg's projected area of the sail and keel is often used to calculate the drag forces. Below we have given an example of how to calculate the drag forces from air and water.

Wind drag

The wind drag is taken to be proportional to the square of the relative wind velocity. $u = U - V$, where U and V are the wind and iceberg velocity vectors in body frame. The wind drag force is given by:

$$F_a = \frac{1}{2} \rho_a C_A A_a |u| u. \quad (3.3)$$

Here A_a is the cross-sectional area of the iceberg above the waterline in a vertical plane normal to the wind, ρ_a is the air density. The air drag coefficient C_A is usually 1 for irregularly shaped objects at high Reynolds numbers.

To be able to give a correct estimate of the wind drag force we need to know the cross-sectional area of the iceberg. It is therefore important to get an accurate estimate of the iceberg's projected area in different directions.

Water drag

The water drag force is modeled in the same way as the wind drag, but the iceberg's underwater area is subdivided into a number of depth layers to account for change in projected area and the Ekman spiral. For each layer, we take the same velocity relative to the iceberg. $w_i = W_i - V$, and again use a quadratic drag law. Also here the velocity is given in the body fixed frame.

Adding all the layers, we get the total drag force:

$$F_w = \frac{1}{2} \rho_w C_W \left| \sum_i A_i |w_i| w_i \right|. \quad (3.4)$$

Where ρ is the density of water and A_i is the cross-sectional area in a vertical plane of the i^{th} layer below the waterline. A

3.2 Iceberg shape geometry and global loads

Timco [1999] presents four important factors when calculating iceberg impact loads on structures:

- The mass of the iceberg for calculation of kinetic energy.
- The length and width of the iceberg for calculation of impact rates and pack ice loads.
- The detailed 3D shape of the iceberg. When an iceberg has an impact against a structure, the force is given by the area of contact and the crushing strength of the ice. The integral of the ice load over the indentation is then equated to the kinetic energy of the iceberg. It is assumed that the indentation increases until the iceberg stops. The highest load during the indentation can be found from the instantaneous area of contact and the pressure from ice failure. Icebergs have been assumed to be spherical or ellipsoidal for impact calculations. For a real iceberg, the contact area may increase more slowly during indentation than it does for a sphere or ellipsoid, thus reducing the impact loads. On the other hand, the contact area of a real iceberg may build up faster than theory, which would increase the impact load.
- The moment of inertia of the iceberg. Most iceberg structure impacts are off center so that the translational kinetic energy is absorbed in both crushings of the ice and rotational kinetic energy. This reduces the ice load compared to a direct impact.

3.3 Iceberg shape and stability

Icebergs, like ships, have six degrees of freedom when floating in the sea: heaving, swaying, surging, rolling, pitching and yawing. We can therefore apply some of the same theory used for ship motions for icebergs with minor modifications. For

instance, the distinction between rolling and pitching is arbitrary for icebergs, both can be considered rolling modes of oscillation.

At sea, wind and waves will excite the iceberg and give it a range of modes of oscillation. We are interested in how these rolling modes can amplify into instability. The iceberg response to these forces will be strongest at the resonant frequencies of its various oscillatory modes. The dynamics of rolling modes for icebergs are calculated in the same way as for ships [Bailey, 1994]. However, we will not focus on dynamic effects on iceberg stability in this thesis. For simplicity, we will assess and predict static stability of an iceberg by using surveys of the iceberg's surface geometry. The calculations will be done based on 2D cross-sections of icebergs. By removing a dimension, the problem is easier to work with. However, we will add a section on how the 2D method can be extended to a 3D method.

Background

The stability of a floating body is determined by the relative movements of its center of gravity and its center of buoyancy as it undergoes small changes in orientation. Figure 3.1 shows a vertical cross-section of a body tilted with an angle θ from its equilibrium position. We can see from the relative position of the upward force of buoyancy and the equal and opposite force of gravity that we will have a restoring moment. The restoring moment will have a lever called GZ. The GZ curve is the curve of GZ against the angle of inclination and is an important for describing the stability of a floating body. From GZ one may derive the moment required to tilt an iceberg to a given angle of heel [Bass and Attwood, 1986].

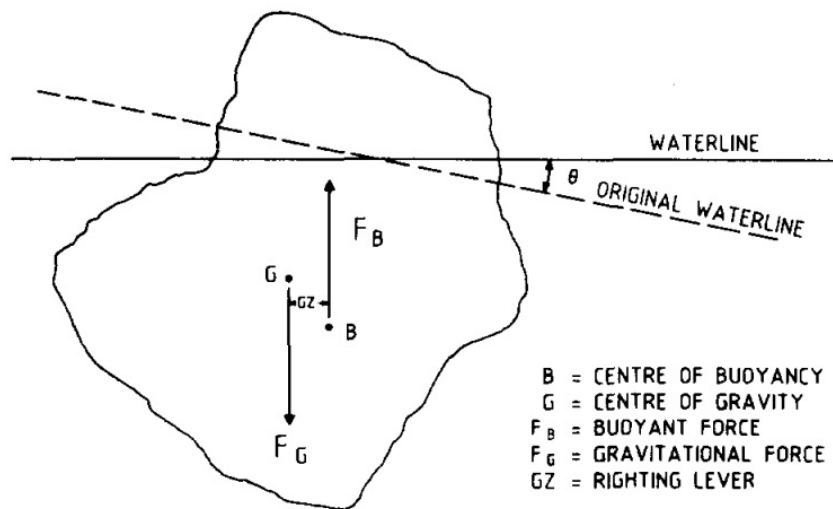


FIGURE 3.1: Iceberg cross-section taken from Bass and Attwood [1986].

For ships, the approximation of GZ is done by GM , which is the linearization of the GZ curve at 0. However, for icebergs this assumption does not hold because of the icebergs irregular shape. Icebergs can experience instability for roll angles close to 0 as seen in Figure 3.2. Records from towing operations also state that icebergs often experienced tow slippage when the icebergs heel more than 10 degrees [Timco, 1999]. This means stability estimations close to zero roll are important, and can often not be estimated by GM due the icebergs irregular shape.

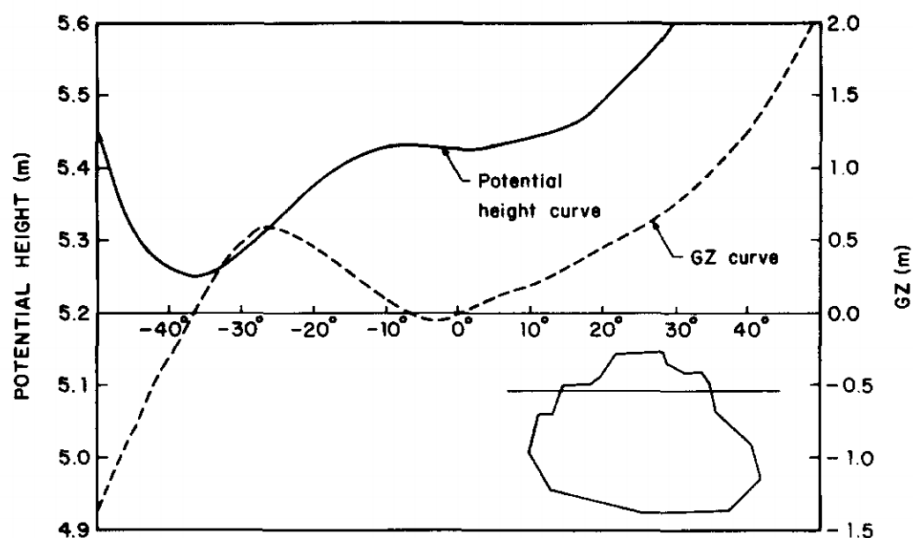


FIGURE 3.2: Iceberg profile data taken from Bass and Attwood [1986].

3.4 An example of how iceberg shape data has been used for stability estimation

An example will be given of how the geometric shape from icebergs can be used for stability estimations. The goal is to estimate how hard we can tow the iceberg before the iceberg rolls over, or the rope slips because the iceberg is tipping too much.

We will now list some assumptions:

- We will calculate on 2D cross-sections of the iceberg, and neglect 3D effects to make the problem simpler. This means we only investigate the stability of the icebergs cross-sections.
- Neglect dynamic effects, and assume the iceberg is in static equilibrium.
- Seawater is set to $1025 \frac{kg}{m^3}$ and ice density to $920 \frac{kg}{m^3}$, where we neglect any variations in salinity.
- The 2D cross-section has its waterline where there is a static equilibrium between the density of water and ice given in the previous bullet point.

Here we will summarize an approach to estimate the stability of an iceberg during a marine operation. In the following sections there will be given a detailed explanation of each bullet point below.

1. Gather relevant data from the iceberg we wish to estimate the stability of. In the example given later on we have used surface length as the parameter to base the estimations on. Surface length is the icebergs largest cross diameter in the waterline.
2. Use a statistical method to extrapolate this data. We have used linear regression to determine a 95 % confidence interval of what the draft is. We have chosen the deepest estimate to be conservative when calculating stability.
3. From the extrapolated data we estimate the keel shape. In this example we create the iceberg keel from the draft estimate. We do this by assuming the keel has a shape of trapezoidal.

4. We then calculate the GZ curve from the estimated iceberg shape.
5. From the GZ value and estimated geometry we can determine the towing force required to roll the iceberg over.

Gather information

First we gather information from the iceberg. We can use computer vision from an UAV to create a digital elevation model (DEM) of the iceberg sail. However, the Arctic is a very hostile environment. UAVs are therefore not always suited. In Section 2.1 we briefly mention some challenges regarding an aerial sensor platform for iceberg surveillance.

Estimation

From the surface data, for example a DEM, the surface volume together with the iceberg density can be used to approximate the volume of the iceberg keel. Also measurements, such as surface length can be used to give information of the keel shape. This is because the surface geometry must satisfy iceberg stability conditions.

Good stability estimations depend on determining the iceberg's distribution of volume. In this example we will predetermine the keel shape as a trapezoid, and scale the trapezoid's size depending on the estimated iceberg draft.

We estimate the iceberg draft (trapezoidal keel height) from using a statistical relations between draft and surface length. The surface length is the longest dimension across the iceberg's surface area, as seen in Figure 3.3.

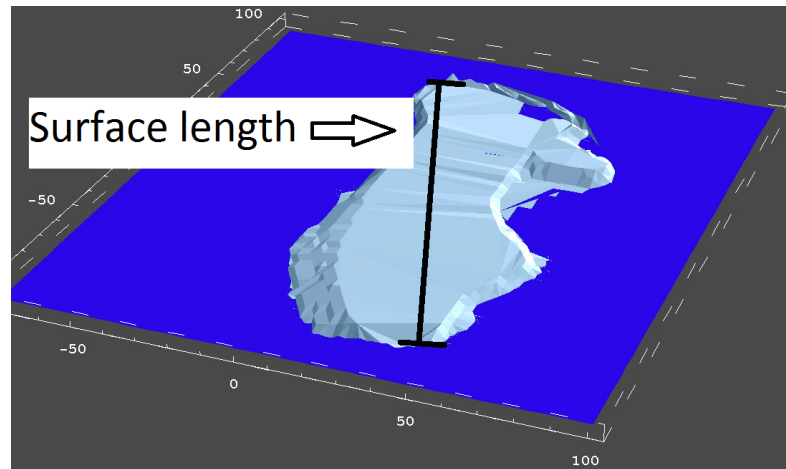


FIGURE 3.3: Surface length used when creating a statistical relation with draft.

We use statistics given by the PERD database taken from the Hibernia field. In Figure 3.4 we see how the draft correlates with the surface length for the Hibernia icebergs.

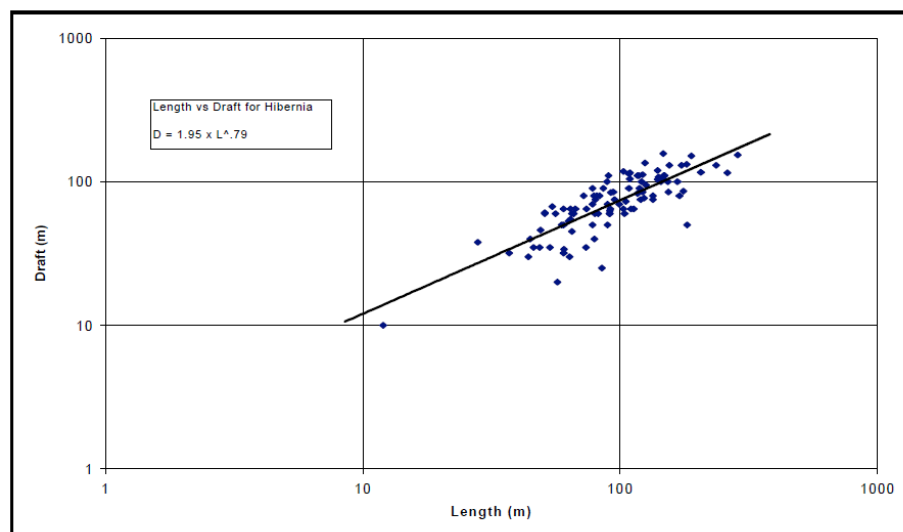


FIGURE 3.4: Length vs. Draft for Hibernia icebergs taken from the [Timco \[1999\]](#).

From the iceberg surface and keel data we have created a 95 % confidence interval to give a conservative estimation of what the iceberg draft is.

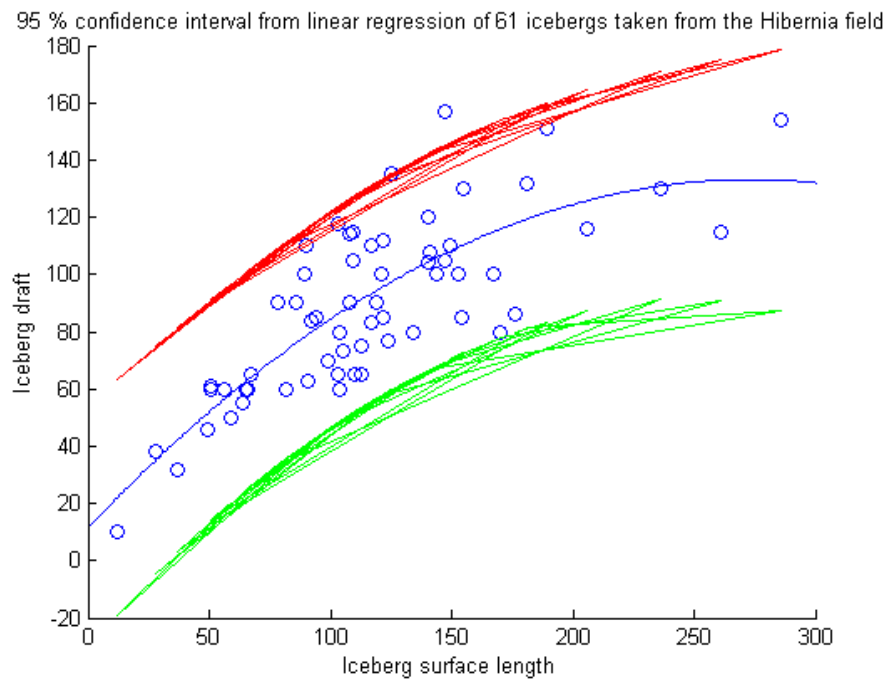


FIGURE 3.5: Made a 95 % confidence interval from iceberg data taken from the Iceberg Profiling Project by the Bedford Institute of Oceanography, 1984/85.

For the 2D model we have estimated the iceberg shape by making a trapezoid from approximating the amount of area beneath the surface and estimating the iceberg draft from the confidence interval. In Figure 3.7, we can see the estimated trapezoids by estimating iceberg drafts from the linear regression model.

The reason for representing the shape by using a trapezoid is because we have enough parameters. To create a more complex shape would require more parameters. In Chapter 5 we will use a neural network to give a more accurate shape estimation.

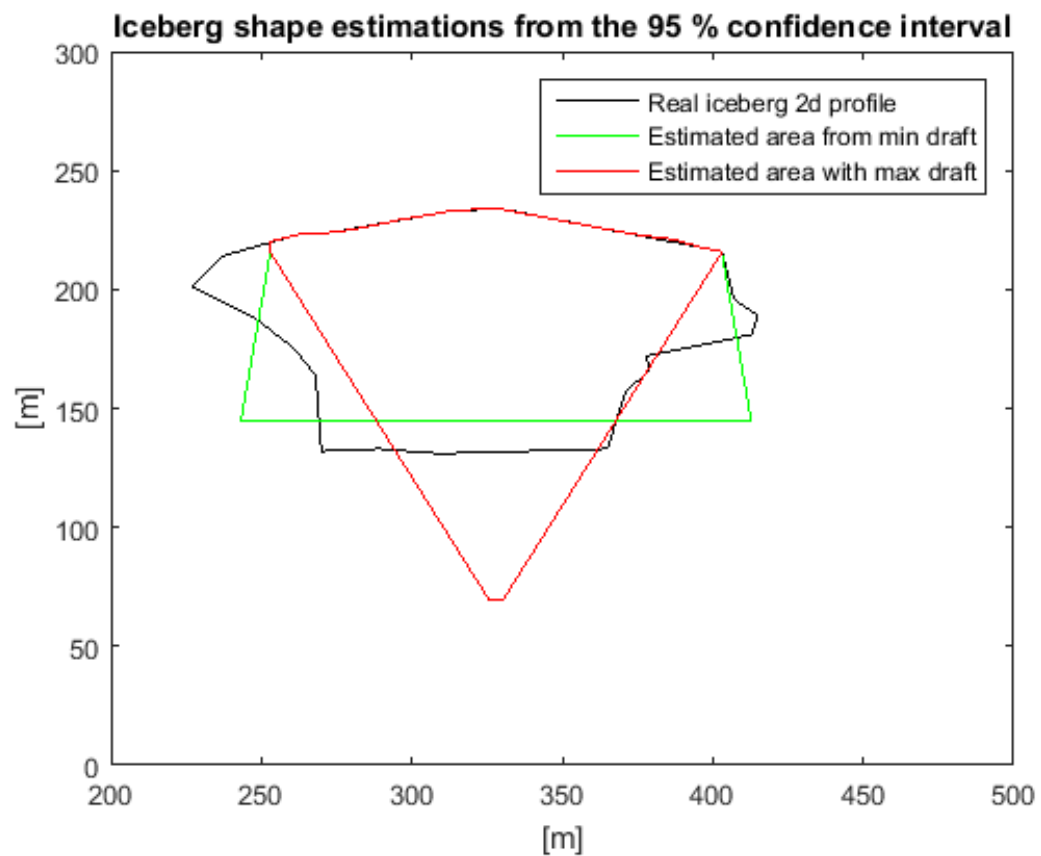


FIGURE 3.6: Here we have created different trapezoidal sizes based on draft variations inside the 95 % confidence interval from iceberg data taken from the Iceberg Profiling Project by the Bedford Institute of Oceanography, 1984/85.

Simulation

From the estimated keel shapes we calculate the iceberg stability at different roll angles. A matlab script has been made to rotate the 2D iceberg profiles and calculates the change in center of buoyancy and creates GZ curves for the different shapes. In Figure 3.7 we can see the GZ curve for when the 2D profile is known and when the profile is estimated. This shows that the iceberg with the greatest draft is the most unstable one.

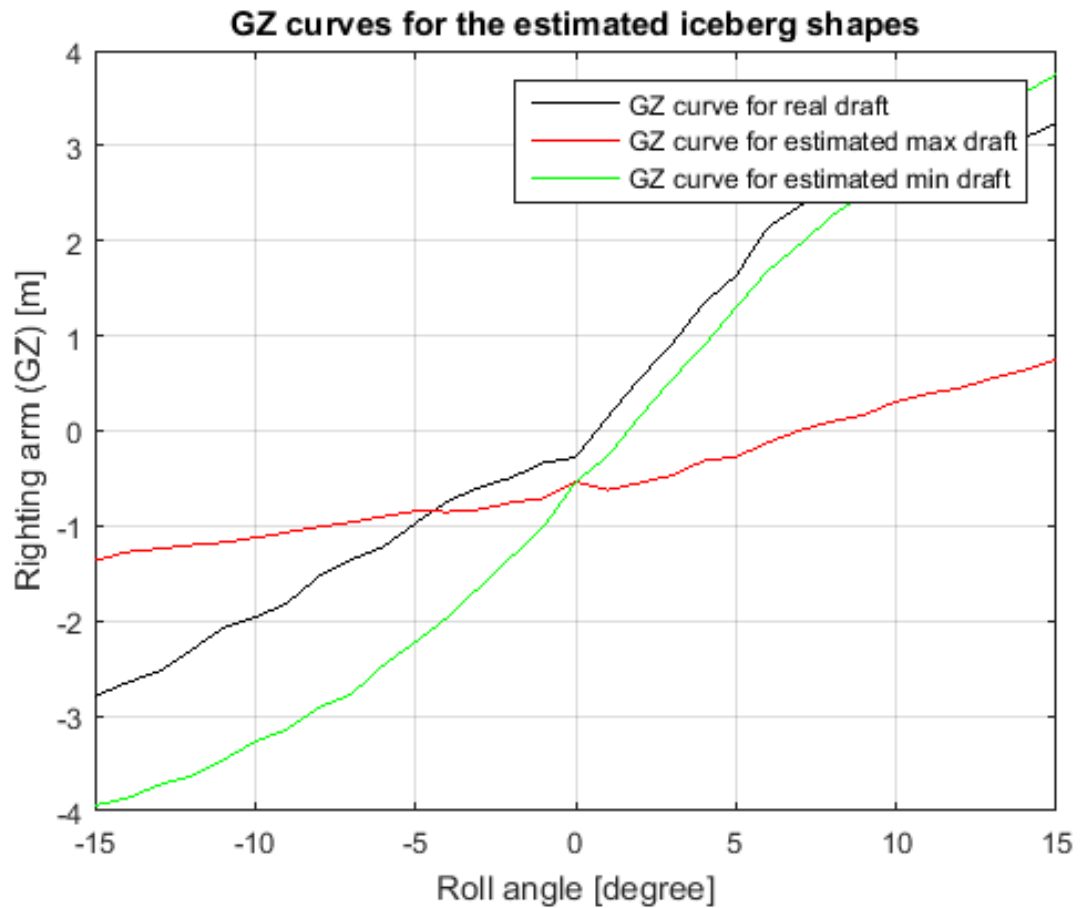


FIGURE 3.7: Here we have a GZ curve for each of the trapezoidal size estimates based on draft variations in the 95 % confidence interval. The iceberg data is taken from the iceberg profiling project by the Bedford Institute of Oceanography, 1984/85.

We also created a polar plot showing which direction the iceberg is most stable. This was done by calculating the second moment of inertia of the iceberg surface area to find the metacentric height (GM). However, this approximation is only valid for small angles, and since icebergs due to their irregular shape, can be unstable for small angles, this approximation is not a good estimate for iceberg stability.

Towing force

From the GZ curves, we can calculate the restoring moment and find out at what speed or force we can tow the iceberg. We can calculate the icebergs restoring moment by:

$$\text{Restoring moment}(\theta) = -\rho g \nabla GZ(\theta). \quad (3.5)$$

Here ∇ is the displacement of the iceberg in cubic meter [m^3], g is gravity and ρ is iceberg density.

The estimated restoring moment can be used during a towing operation to avoid the iceberg's angle of inclination becoming too large, in worst case resulting in tow slippage or rolling. The limiting restoring moment can be used in a dynamic model to find the speed at which the iceberg can be towed.

3.5 Discussion

In terms of predicting real iceberg stability, the dynamic effects need to be included. Bailey [1994] proposes a way to predict towing force based on rolling frequency f_t of towing torque T_T . Also by attaching an inertial measurement unit on the iceberg and measuring the frequency of excitation would give more exact stability estimations.

We initially assumed that larger draft would make the iceberg more unstable. However, in Figure 3.7 we see that the blue GZ curve for the real iceberg 2D profile is more stable than the green estimated iceberg GZ curve that has less draft. This is because the shape close to the waterline has a larger effect on stability for small perturbations than the iceberg draft. This shows that the distribution of volume close to the surface is critical when estimating iceberg stability. The trapezoid keel shape does not represent the iceberg shape accurately. The sudden slope transition from iceberg sail to the keel with the trapezoid is non-physical and does not exist for real icebergs. This is most likely because of how icebergs are made, and how they eventually get rounder from rolling and changing their orientation within their lifespan, before melting.

In the example we used a surface length from 2D iceberg profiles and estimated the draft from a confidence interval. However, the confidence interval was based on 3D icebergs and not 2D icebergs. So the confidence interval we used most likely overestimated the iceberg draft.

3D analysis

One approach to convert the 2D example to 3D would be to add a dimension by multiplying the trapezoid with a width, making it a 3D trapezoid. We do this for several cross-sections along the iceberg so that we create many thin 3D icebergs. The thin 3D icebergs can then be added together creating the total iceberg keel geometry.

The procedure would be to measure many surface lengths as seen in Figure 3.8. Then we can estimate the draft based on the surface length and create a 2D trapezoidal keel approximation for that cross-section. The trapezoid can then be elongated to the next cross-sectional measurement, so that it becomes 3D. Finally we can add up all the thin 3D keel estimations to a total. In (3.6) we have shown how we can find the restoring moment from finding the GZ curve and displacement from each trapezoid:

$$\text{Restoring moment iceberg}(\theta) = -\rho g \sum_{i=1}^N GZ(\theta)_i \nabla_{\text{trapezoid},i}(\theta). \quad (3.6)$$

In Figure 3.8 we can see how the surface length $\mathbf{b2}$ can be used for draft estimation of \mathbf{h} . We can then do this for several surface lengths and add all the trapezoids with a width \mathbf{L} .

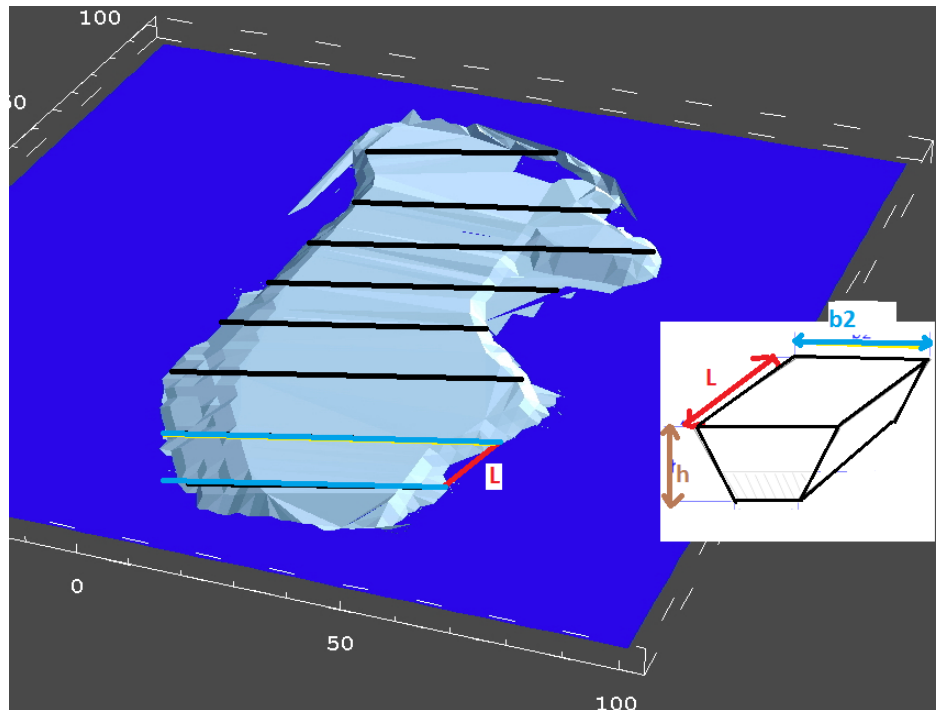


FIGURE 3.8: How we can approximate 3D stability from 2D profile estimations in one roll direction.

The 3D estimation will have the same problem as for the 2D case with estimating the distribution of volume close to the surface. Also by choosing a trapezoid shape, and elongating it, we will give an uneven 3D distribution of volume, which will give an inaccurate stability estimation in different directions.

In Figure 3.9 we have made an illustration of how to use the 2D trapezoidal estimations made earlier for a 3D stability calculation.

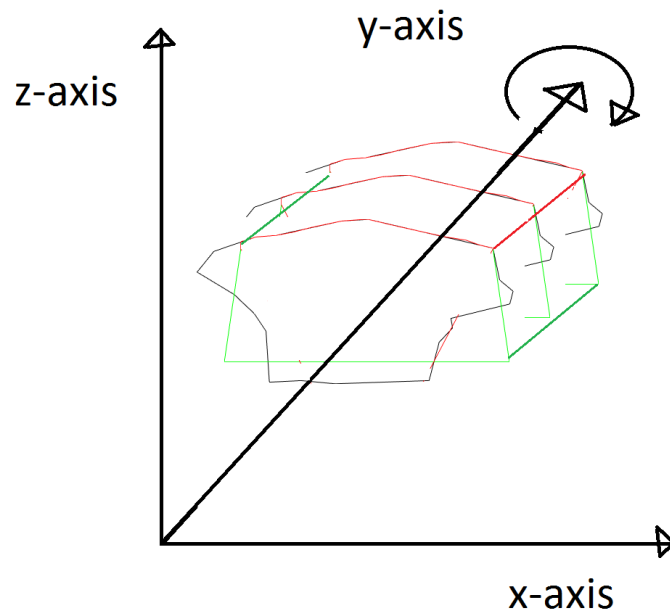


FIGURE 3.9: How we can approximate 3D stability from 2D profile estimations in one roll direction.

It is important to note that even by obtaining a more accurate estimation technique for iceberg keel geometry, the effect of the iceberg's in-homogeneous density distribution could still cause the predicted stability to be inaccurate.

Chapter 4

Theoretical Background

This chapter will provide a theoretical background for modeling an artificial neural network (ANN) for function approximation. This is because we will design a feedforward network for estimating iceberg keel geometry in the next chapter. A feedforward network is a subset of the class of non-linear regression. The motivation for using a neural network instead of other statistical tools can be summarized as:

- They are useful for situations where there is a lot of data, but little underlying theory to describe the model. For icebergs, because of their irregular shape, it is hard to model the connection between the sail profile and keel. Artificial networks can find and use weak non-linear relations between data without any prior assumptions about the problem space.
- Artificial neural networks are adaptive. The network can continuously be trained with new information, and improve its estimations.
- Artificial networks are good at self-organizing information, and are relatively simple to construct for data sets with many parameters.

A general description of a neural network is a machine that is designed to model the way the brain performs particular tasks or functions [[Haykin, 2001](#)]. The network we will use is an artificial neural network for machine learning. This is a family of statistical learning algorithms inspired by the biological neural networks.

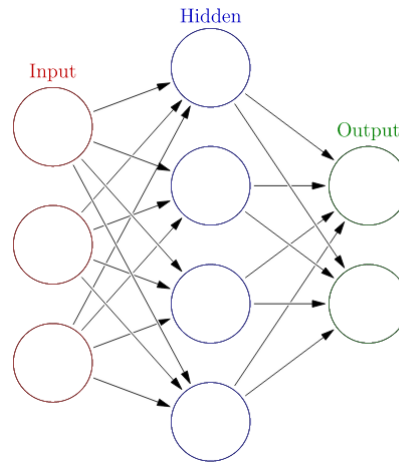


FIGURE 4.1: Conceptual drawing of an artificial neural network [Gos].

In Figure 4.1 we show a simple illustration of a 3 layered artificial neural network.

The layer between the input and output layer has no interaction with the environment and is known as the hidden layer. By adding more hidden layers we increase the complexity of an ANN, and thus its computational capacity. The added complexity will also increase the time to train the network. The final layer that interacts with the output is known as the output layer.

Artificial neural networks are presented as systems of interconnected neurons that send signals to each other. The connection between neurons can have numeric weights that get tuned based on experience. This makes neural networks adaptive to inputs. A neural network has commonly two characteristics:

- They consist of a set of adaptive weights where the numerical weights are tuned by learning algorithms.
- They can approximate non-linear functions of their inputs.

4.1 Neuron model

As mentioned, the ANN consists of many neurons. These neurons can have one or multiple inputs. To begin with, we can explain how a multi-input neuron works as seen in Figure 4.2.

In Figure 4.2 we have a multiple input neuron. We call the multiple inputs for a vector \mathbf{p} , where the individual inputs are given as p_1, p_2, \dots, p_R . This vector is multiplied by a weight matrix \mathbf{W} , consisting of individual weights, $w_{1,1}, w_{1,2}, \dots, w_{1,R}$. The first indice of the elements in the weight matrix indicates the particular neuron destination for the weight. The second index indicates the source from where the signal is being transmitted to the neuron.

The product of \mathbf{W} and \mathbf{p} is a vector, \mathbf{Wp} , where a bias \mathbf{b} is added before this term is sent to the summer. The summer output is n , and is often called "the net input."

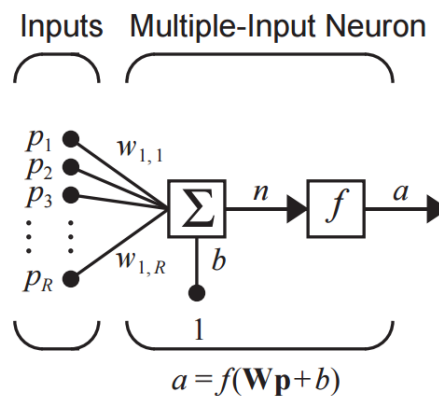


FIGURE 4.2: Multiple-Input Neuron from Hagan et al. [2002].

The net input n , can be calculated by:

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b. \quad (4.1)$$

In matrix form the expression can be written as:

$$\mathbf{n} = \mathbf{Wp} + \mathbf{b}. \quad (4.2)$$

This net input is then sent to a transfer function \mathbf{f} , which can produce the output vector \mathbf{a} . The neuron output can be written as:

$$\mathbf{a} = \mathbf{f}(\mathbf{Wp} + \mathbf{b}). \quad (4.3)$$

The output depends on which transfer function we chose. We will discuss transfer functions in the next section.

4.2 Transfer functions

In Figure 4.2 we can see the transfer function, also known as an activation function, shown by f . The transfer function may be a linear or a non-linear function of the input \mathbf{n} depending on what problem we want to solve with the artificial neural network.

We use these functions in the hidden layer to introduce non-linearity into the network. This is because a linear function of a linear function is again a linear function. This capability to represent non-linear functions is what makes multi-layered networks very powerful. [Hornik, Stinchcombe, and White \[1989\]](#) have shown that a two-layered network, with a tan-sigmoid transfer function in the hidden layer and a linear transfer function in the output layer, can approximate any function of interest. This can be done with any degree of accuracy as long as the network has enough neurons.

In the model presented in Chapter 6, we have chosen a tan-sigmoid function as the non-linear transfer function and a linear purelin transfer function in the output layer. This is one of the most common choices. We could almost use all types of non-linear functions, but because of back-propagation learning the transfer function must be differentiable.

The sigmoid functions are also easier to train than for example threshold units. This is because a small change in the weights will usually produce an equal relative change in the outputs. While for a threshold unit a slight change in the weights will often not produce a change in output. The tan-sigmoid function given by:

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (4.4)$$

4.3 Network architecture

One neuron is often not sufficient when dealing with many inputs. We can solve this by having several neurons operating in parallel. We call this a "layer" of neurons [Hagan et al., 2002].

As mentioned, ANNs have three types of layers: input, hidden, and output. When creating the neural network architecture we need to come up with values for the number of layers of each type and also the number of nodes in these layers. However, the exact number of hidden layers S^1 to use is something that can not be known before training. Later on when designing the network for a particular case, we will check the performance of the network for different numbers of neurons.

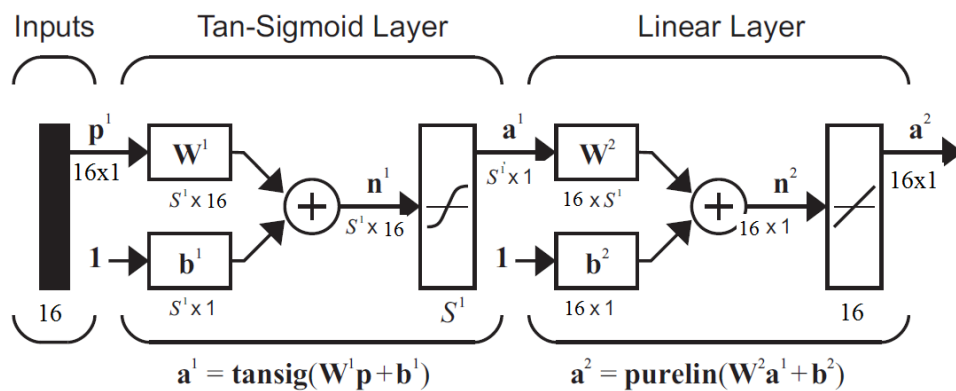


FIGURE 4.3: Illustration of network Architecture.

We have decided to only have one hidden layer for the network, to make it as simple as possible. This will prevent having too many parameters to estimate, especially given the low iceberg sample size.

4.4 Training the network

Here we will shortly introduce the learning technique that will be used later on. It is called performance learning, which is one of several different classes of network learning laws. In performance learning the network parameters (weights and biases) are adjusted to optimize the performance of the network.

Back-propagation

In 1969, [Minsky \[1969\]](#) presented a two-layered feed-forward network, but they did not find a solution to the problem of how to adjust the weights and biases in the network. A solution to this problem was presented by [Williams and Hinton \[1986\]](#), [Parker \[1985\]](#) and [Cun \[1985\]](#). The idea was to use the errors from the output layers to back-propagate them and use it to determine the units in the hidden layers. This is why the algorithm is called the back-propagation learning rule.

The following section will explain the back-propagation algorithm. The explanation is taken from [Hagan et al. \[2002\]](#).

Forward Propagation

As described earlier, the output from one layer is the input to the next layer. We can calculate the final output \mathbf{a}^M from the following equations. \mathbf{M} is the number of layers in the network.

$$\mathbf{a}^{m+1} = \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \quad \text{for } m = 0, 1, \dots, M - 1 \quad (4.5)$$

The neurons in the first layer receive external input matrix \mathbf{P} :

$$\mathbf{a}^0 = \mathbf{p}, \quad (4.6)$$

this provides the starting point for equation (4.5). The outputs of the neurons in the last layer are considered the network outputs.

$$\mathbf{a} = \mathbf{a}^M \quad (4.7)$$

Performance Index

The back-propagation algorithm uses the mean square error as its performance index. The performance index will later on be used to optimize the weights and biases of the ANN.

Each input matrix \mathbf{p} to a network, has a target output matrix \mathbf{t} . We find the performance index by calculating the forward propagation matrix \mathbf{a} , from a corresponding input matrix \mathbf{p} . We can then calculate the mean square error between the forward propagation matrix \mathbf{a} , and the target output matrix \mathbf{t} . The back-propagation algorithm wants to adjust the parameters in order to minimize the mean square error, $F(\mathbf{x})$:

$$F(\mathbf{x}) = E[e^2] = E[(t - a)^2], \quad (4.8)$$

where \mathbf{x} is the vector of network weights and biases. If the network has multiple outputs, the expression becomes:

$$F(\mathbf{x}) = E[\mathbf{e}^T \mathbf{e}] = E[(\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a})]. \quad (4.9)$$

We will approximate the mean square error by:

$$\hat{F}(\mathbf{x}) = (\mathbf{t}(k) - \mathbf{a}(k))^T (\mathbf{t}(k) - \mathbf{a}(k)) = \mathbf{e}^T(k) \mathbf{e}(k), \quad (4.10)$$

where the expectation of the square error has been replaced by the squared error at iteration k .

Levenberg-Marquardt algorithm

In this section, we will present the back-propagation method used later on. There are several ways of minimizing the error and optimizing the weights and biases of the network. However, some methods provide faster convergence than others. We will use a back-propagation algorithm called Levenberg-Marquardt. It is based on Newton's method and is a standard numerical optimization technique.

The goal is to optimizing the performance index $\mathbf{F}(\mathbf{x})$. Where \mathbf{x} is the vector of network weights and biases. We will use Newton's method for this:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{A}_k^{-1} \mathbf{g}_k. \quad (4.11)$$

where $\mathbf{A}_k \equiv \nabla^2 \mathbf{F}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$ and $\mathbf{g}_k \equiv \nabla \mathbf{F}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_k}$.

If we assume $F(\mathbf{x})$ is a sum of squares function:

$$F(\mathbf{x}) = \sum_{i=1}^N v_i^2(\mathbf{x}) = \mathbf{v}^T(\mathbf{x})\mathbf{v}(\mathbf{x}). \quad (4.12)$$

then the j th element of the gradient would be:

$$[\nabla F(x)]_j = \frac{\partial F(\mathbf{x})}{\partial x_j} = 2 \sum_{i=1}^N v_i(\mathbf{x}) \frac{\partial v_i(\mathbf{x})}{\partial x_j}. \quad (4.13)$$

The gradient can therefore be written in the matrix form:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{v}(\mathbf{x}), \quad (4.14)$$

where:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \cdots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (4.15)$$

,

is the Jacobian matrix.

Now we want to find the Hessian matrix. The k,j element of the Hessian matrix are:

$$[\nabla^2 \mathbf{F}(\mathbf{x})]_{k,j} = \frac{\partial^2 \mathbf{F}(\mathbf{x})}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left\{ \frac{\partial v_i(\mathbf{x})}{\partial x_j} \frac{\partial v_i(\mathbf{x})}{\partial x_k} + v_i(\mathbf{x}) \frac{\partial^2 v_i(\mathbf{x})}{\partial x_k \partial x_j} \right\}. \quad (4.16)$$

In matrix form the Hessian can be expressed as:

$$\nabla^2 \mathbf{F}(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\mathbf{S}(\mathbf{x}), \quad (4.17)$$

where:

$$\mathbf{S}(\mathbf{x}) = \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}). \quad (4.18)$$

If we assume that $\mathbf{S}(\mathbf{x})$ is small, we can approximate the Hessian matrix as:

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}). \quad (4.19)$$

If we then substitute (4.19) and (4.14) into (4.11) we obtain the Gauss-Newton method:

$$\Delta x_k = x_{k+1} - x_k = [\mathbf{J}^t(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k). \quad (4.20)$$

The advantage of Gauss-Newton over the standard Newtons method is that we do not need to calculate any second derivatives. However, a problem with the Gauss-Newton method is that the matrix $\mathbf{H} = \mathbf{J}^T\mathbf{J}$ may not be invertible. We can solve this by modifying the approximated Hessian matrix:

$$\mathbf{G} = \mathbf{H} + \mu\mathbf{I}. \quad (4.21)$$

To see how this matrix can be made invertible, we suppose that the eigenvalues and eigenvectors of \mathbf{H} are $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ and $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n\}$. Then

$$\mathbf{G}\mathbf{z}_i = [\mathbf{H}\mu + \mathbf{I}]\mathbf{z}_i = \mathbf{H}\mathbf{z}_i + \mu\mathbf{z}_i = \lambda_i\mathbf{z}_i + \mu\mathbf{z}_i = (\lambda_i + \mu)\mathbf{z}_i. \quad (4.22)$$

The eigenvectors of \mathbf{G} are the same as the eigenvectors of \mathbf{H} , and the eigenvalues of \mathbf{G} are $(\lambda_i + \mu)$. \mathbf{G} can be made positive definite by increasing μ until $(\lambda_i + \mu) > 0$ for all i , and therefore the matrix will be invertible.

By adding this change to the Hessian matrix, we get the *Levenberg-Marquardt algorithm*:

$$x_{k+1} = x_k - [\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k\mathbf{I}]^{-1}\mathbf{J}^T(\mathbf{x}_k)\mathbf{v}(\mathbf{x}_k). \quad (4.23)$$

We will not show how we calculate the Jacobian for this algorithm. See [Hagan et al. \[2002\]](#) Chapter 13 for a detailed description.

4.5 Generalization

A major advantage with neural networks is their ability to generalize. Generalizing is when a trained network can classify data it has never seen before. The network should perform well on both the training data and on new data. This concept is called generalization [Hagan et al., 2002]. We want to design a simple network to reduce the possibility of errors. We achieve this by creating a network that contains the smallest number of free parameters, in the form of weights and biases. This is done by reducing the number of neurons in the model.

To be able to evaluate generalization when designing a neural network we split the data into three categories:

- Training samples.
- Validation samples.
- Test samples.

The training set is used to update the weights and biases of the network to minimize the mean square error between the target outputs and the forward propagated outputs. We use the validation set to see how well the network performs on new data. The training should be stopped in the minimum of the validation set error. At this point, the network generalizes best. If learning is not stopped the network will overfit, which will decrease its performance on the new data. Finally, when the network is finished with training, we can test the performance on the third data set, the test set.

Given that the amount of data was unlimited, which means that the number of data points is significantly larger than the number of network parameters, we would not have a problem with overfitting. But because the training data we use later on is limited we must investigate how we can improve generalization.

How to improve generalization?

We can improve generalization by either restricting the number of weights or limiting the magnitude of the weights. There are several ways of improving generalization. We will look at an algorithm that can do this without changing the

number of neurons. We will use a method called regularization that modifies the performance index to include a term that penalizes network complexity.

When we increase the weights, the slope of the network function will also increase. This means that when the weights are large, the functions created by the network will have large slopes. This makes the network more likely to overfit. However, if we can restrict the weights, then the network function will create a smoother interpolation.

With every network we start with a training set consisting of inputs and corresponding targets given by:

$$\mathbf{p}_1, \mathbf{t}_1, \mathbf{p}_2, \mathbf{t}_2, \dots, \mathbf{p}_Q, \mathbf{t}_Q, \quad (4.24)$$

where Q represents the number of inputs.

For the generalization calculations, we will assume that the target outputs are generated by inputs with:

$$\mathbf{t}_q = \mathbf{g}(\mathbf{p}_q) + \epsilon_q. \quad (4.25)$$

Here $\mathbf{g}(\cdot)$ is a unknown function, and ϵ_q is a zero mean random variable white noise. The objective when training the neural network is to approximate the function $\mathbf{g}(\cdot)$ while ignoring the noise, ϵ_q .

In 4.26 we can see the regularization term written as the sum of squares of the networks weights. The ratio $\frac{\alpha}{\beta}$ determines the complexity of the network.

$$F(\mathbf{x}) = \beta \sum_{q=1}^Q (\mathbf{t} - \mathbf{a})^T (\mathbf{t} - \mathbf{a}) + \alpha \sum_{i=1}^n x_i^2. \quad (4.26)$$

How well the network generalizes is determined by the choice α and β . We will use Bayesian analysis to estimate α and β . This method is called Bayesian regularization.

The method we will present is from MacKay [1992].

We first assume that the networks weights are random variables and then create a probability function using Bayes rule. We chose the weights that maximize the conditional probability of the weights given the data.

$$P(\mathbf{x}|\mathbf{D}, \alpha, \beta, \mathbf{M}) = \frac{P(\mathbf{D}|\beta, \mathbf{M})P(\mathbf{x}|\alpha, \mathbf{M})}{P(\mathbf{D}|\alpha, \beta, \mathbf{M})}. \quad (4.27)$$

Here \mathbf{x} contains all the weights and biases in the network. \mathbf{D} represents the training data set and \mathbf{M} describes the network model by how many layers, and neurons are in each layer.

The probability density function $P(\mathbf{D}|\beta, \mathbf{M})$ represents the probability for the data given, the set of weights \mathbf{x} , the parameter β , and the choice of network \mathbf{M} . By assuming the noise term in 4.25 is independent and Gaussian distributed. We can write the density function as:

$$P(\mathbf{D}|\beta, \mathbf{M}) = \frac{1}{Z_D(\beta)} \exp(-\beta E_D), \quad (4.28)$$

here $\beta = \frac{1}{2\sigma_s^2}$, where σ_s^2 is the variance of each element of ϵ . And

$$Z_D(\beta) = (2\pi\sigma_s^2)^{\frac{N}{2}} = \left(\frac{\pi}{\beta}\right)^{\frac{N}{2}}, \quad (4.29)$$

where N is $Q \times S^M$. Q is the number of inputs and M is the number of neuron layers. The second term in the 4.29 allows us to incorporate prior knowledge through its density function.

$$P(\mathbf{x}|\alpha, \mathbf{M}) = \frac{1}{Z_W(\alpha)} \exp(-\alpha E_W), \quad (4.30)$$

where $\alpha = \frac{1}{2\sigma_W^2}$, and σ_W^2 is the variance of each weight. E_w is the sum squared weights and:

$$Z_W(\alpha) = (2\pi\sigma_W^2)^{\frac{n}{2}} = \left(\frac{\pi}{\alpha}\right)^{\frac{n}{2}}, \quad (4.31)$$

where n is the number of weights and biases in the network.

We do not need to be concerned with $P(\mathbf{D}|\alpha, \beta, \mathbf{M})$, because it is not a function of \mathbf{x} and is a normalizing factor. We can now write (4.27) as:

$$\begin{aligned} P(\mathbf{x}|\mathbf{D}, \alpha, \beta, \mathbf{M}) &= \frac{\frac{1}{Z_W(\alpha)} \frac{1}{Z_D(\beta)} \exp(-(\beta E_D + \alpha E_W))}{\text{NormalizingFactor}} \\ &= \frac{1}{Z_F(\alpha, \beta)} \exp(-F(\mathbf{x})). \end{aligned} \quad (4.32)$$

To find the most probable values of the weights we should maximize the posterior density $P(\mathbf{x}|\mathbf{D}, \alpha, \beta, \mathbf{M})$, which is equivalent to minimizing the regularized performance index $F(\mathbf{x}) = \beta E_D + \alpha E_W$. We will mark the weights that maximize the posterior density function as \mathbf{x}^{MP} .

To see how we can find the unknown parameters, see [Hagan et al. \[2002\]](#), Chapter 13.

Chapter 5

Case study: Estimation of Keel Geometry from Surface Data

Problem statement: When performing marine operations in the Arctic, geometry and size of an iceberg is necessary for calculating the icebergs trajectory and stability. However, obtaining this data is not easy. Today we can use underwater vehicles to scan the iceberg's keel using sonar and then create 3D models from this data. That said, operating underwater vehicles as a sensor platform demands a lot of resources. We want to find a way to obtain accurate estimates of the iceberg keel geometry without relying on an underwater sensor platform.

Proposed solution: The solution is to use iceberg sail data to estimate the keel geometry. We will therefore not rely on an underwater sensor platform. The iceberg sail data can be estimated using an aerial sensor platform instead. However, this method does need pre-existing data of iceberg sails and their corresponding keels. Therefore the underwater sensor platform is required, but only for a limited amount of time, until the data sets are good enough to build a statistical basis for estimation.

When iceberg keel and sail data is obtained we can use a statistical tool for the estimation. In this case we will build an artificial neural network. The goal is that the neural network can generalize non-linear relationships between iceberg surface shape inputs and iceberg subsurface outputs. We hope that this method will be a part of the next generation ice management system.

We will now explain how we prepare the iceberg data and select parameters for the network.

5.1 Data collection and preprocessing

For this case 2D cross-sections of icebergs will be the statistical basis for the neural network. Later on there will be a section on how the method can be extended to a 3D case. Horizontal stereo cameras were used to obtain the 2D above water dimensions of the icebergs, and a vertically lowered sidescan sonar was used to obtain the iceberg draft and vertical profile of the iceberg's underwater shape. We have 42 2D profiles of iceberg sails and their corresponding keels from the PERD database. The 2D profiles have been extracted from scanned images. In Figure 5.1 we see an iceberg profile before extraction.

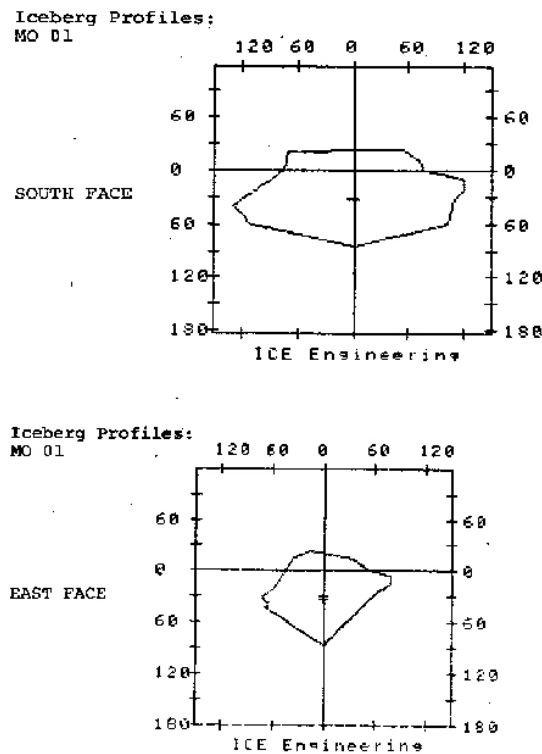


FIGURE 5.1: Example of a 2D cross-section used from the PERD database.

We detect and extract the outline of the iceberg 2D profile into coordinates. In Figure 5.2 the detected outline is shown.

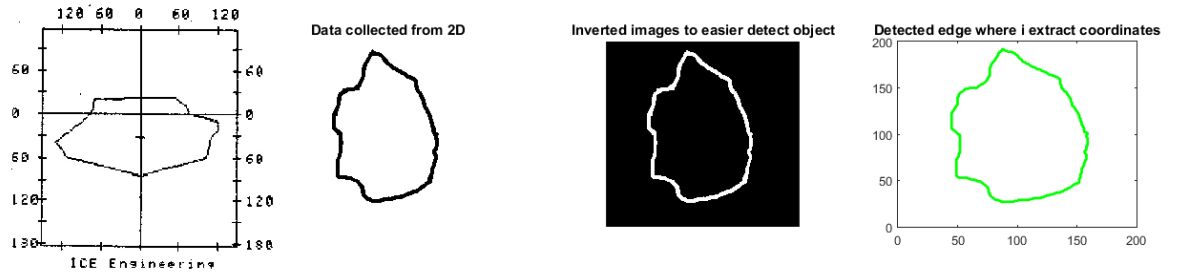


FIGURE 5.2: Image preprocessing of the 2D profiles from the PERD database.

For the 2D cross-section far left in Figure 5.2 the waterline is marked with a black line. This waterline is the iceberg's global waterline in 3D. For this case, we will move the global black waterline for the cross-sections. This is so that the cross-sectional area ratio satisfies Archimedes law. In (5.1) to (5.5) we can see how the keel and sail volume ratio can be found. Where F_b is the buoyancy force of the iceberg. F_g is the gravity force of the iceberg. ∇_{berg} represents the displaced volume of the iceberg and V_{ice} represents the total iceberg volume.

$$\sum F = 0 \quad (5.1)$$

$$F_b = F_g \quad (5.2)$$

$$\nabla_{berg} \times \rho_{water} = V_{ice} \times \rho_{ice} \quad (5.3)$$

$$\frac{\nabla_{berg}}{V_{ice}} = \frac{\rho_{ice}}{\rho_{water}} \quad (5.4)$$

$$(5.5)$$

When choosing a constant density ratio we have neglected variations in salinity in both seawater and sea ice.

After extracting the outline coordinates they need to be reduced before being used as input values for the neural network. We do this by using a Matlab code by [Bone \[2014\]](#) which reduces the number of vertices in a closed polygon. It does this by calculating the importance of each vertex based on angle and segment length and then removing the least important one. The process is repeated until the desirable

number of vertices is reached. For the neural network we have reduced the surface and subsurface polygon to only 8 vertices. By reducing the number of vertices even more, we would lose too much of the original shape. The MatLab program can be found uploaded into a folder in DAIM. In Figure 5.2 we can see this transformation.

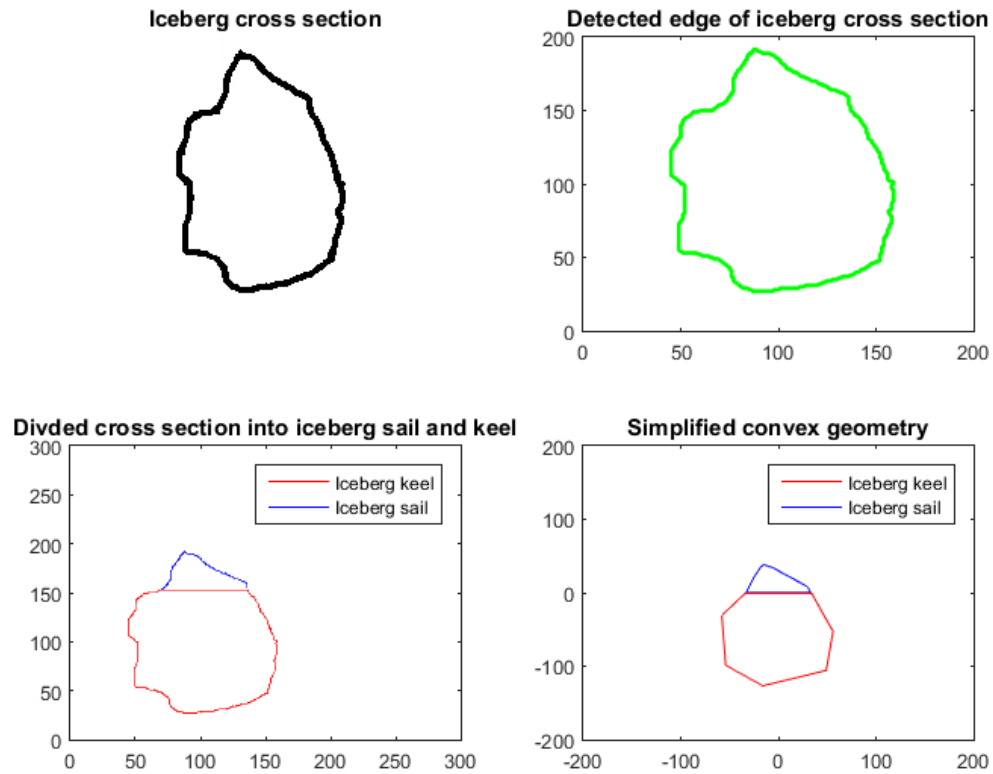


FIGURE 5.3: Preprocessing of iceberg data.

5.2 Selecting the architecture

Neural networks have 3 types of layers: input, hidden, and output. When creating the neural network architecture we need to come up with values for the number of layers of each type and also the number of nodes in these layers. However, the number of hidden layers S^1 is something that cannot be known before training. In the next section we will come back to this.

For the network a tan-sigmoid transfer function will be used in the hidden layer, and a linear output layer. This is a typical network for function approximation. In Figure 5.4 we see the network architecture for the neural network.

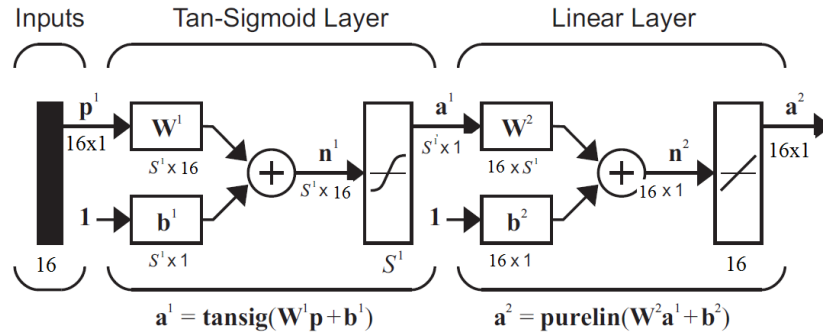


FIGURE 5.4: Illustration of network Architecture.

5.3 Training the network

The objective when training a neural network is to make it perform well on both the training data, and new data. This concept is called generalization [Hagan et al., 2002]. We also want to design a simple network. This means a model that contains the smallest number of free parameters (weights and biases), which means reducing the number of neurons in the model. Also by avoiding complexity we reduce the possibility of errors and the time used to train the network.

Input matrix :

$$P = [16 \times 42]. \tag{5.6}$$

Target matrix:

$$T = [16 \times 42]. \tag{5.7}$$

We have divided the image data into three sample sets:

- Training samples 70 % (30 samples).
- Validation samples 15 % (6 samples).
- Test samples 15 % (6 samples).

(See section 4.5 for what the difference sets are for.)

First we trained the network using the Levenberg-Marquardt algorithm for different network sizes. We created a network with 3, 5 and 10 nodes where we

calculated the mean square error and the regression value. If the number of neurons are too many, the network might overfit the training data.

The mean square error is the average squared difference between outputs and targets. Lower values are better, zero means no error. The regression value R measures the correlation between outputs and targets. R equal to 1 means a close relationship, and 0 means a random relationship. R can be calculated from (5.8).

$$R(a, T) = \frac{Cov(a, T)}{\sqrt{Cov(a, T)Cov(a, T)}} \quad (5.8)$$

The mean square error (MSE) and regression values (R) for the different networks are found in Table 5.1, 5.2 and 5.3:

	Samples [T]	MSE [-]	R [-]
Training	30	0.0194	0.932
Validation	6	0.0258	0.914
Testing	6	0.0273	0.907

TABLE 5.1: Levenberg Marquardt backpropagation algorithm of network with 3 hidden neurons.

	Samples [T]	MSE [-]	R [-]
Training	30	0.0181	0.94
Validation	6	0.0364	0.876
Testing	6	0.0162	0.941

TABLE 5.2: Levenberg Marquardt backpropagation algorithm of network with 5 hidden neurons.

	Samples [T]	MSE [-]	R [-]
Training	30	0.0210	0.932
Validation	6	0.0468	0.831
Testing	6	0.0373	0.871

TABLE 5.3: Levenberg Marquardt backpropagation algorithm of network with 10 hidden neurons.

The errors from all three networks are similar. The network with 5 nodes gives the best result and can be seen in Figure 5.5. A mean square error of 0.0162 from the target samples and produced output seems high given that we have scaled the data between 0 and 1.

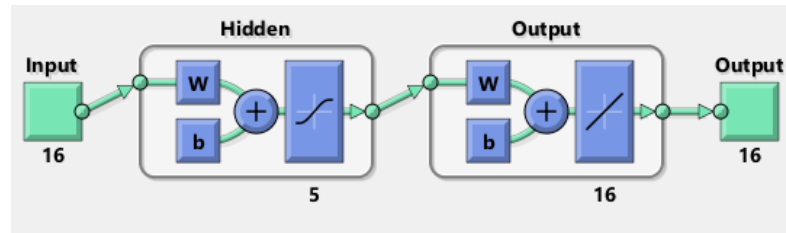


FIGURE 5.5: 5 node network with 16 inputs (x,y -coordinates) and 16 targets (x,y -coordinates) from Matlab.

In Figure 5.6 we can see the MSE plotted against number of epochs. An epoch is a measure of the number of times all of the training vectors are used once to update the weights. After only two epochs, we can get the best validation performance, which is the best performance on new data. By stopping the network when we reach the best performance on validation samples, the network will perform well on new data without overfitting.

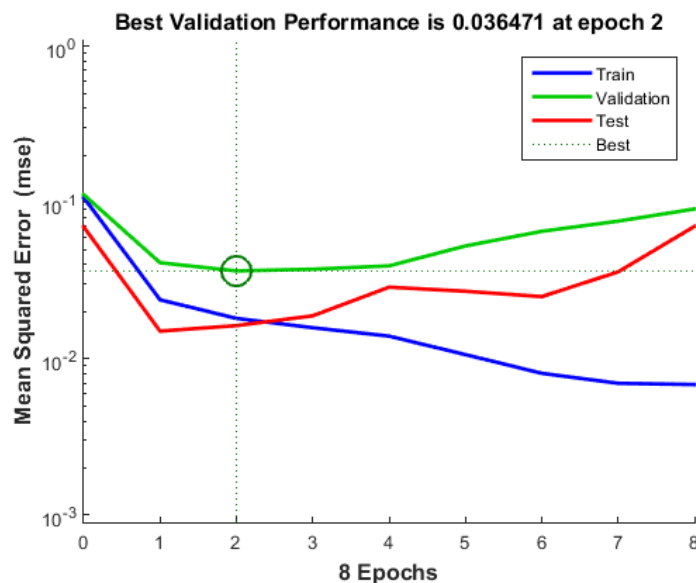


FIGURE 5.6: Performance plot of the training, validation and test set using the Levenberg-Marquardt algorithm.

In Figure 5.7 we see the distribution of error in a histogram for each output for a 10 node network using the Levenberg-Marquardt training algorithm. We see that most sample outputs have close to 0 error from the targets. However, we have some outliers, which give a bad shape estimation of the iceberg.

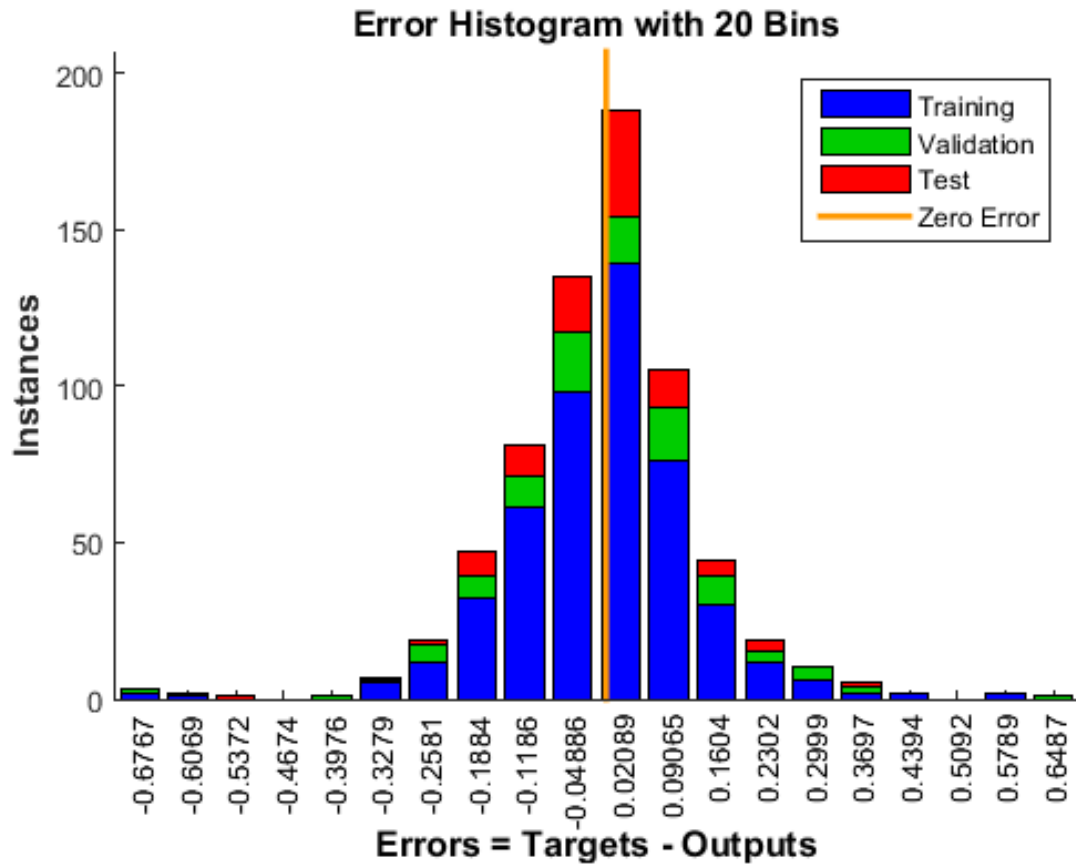


FIGURE 5.7: Histogram plot of errors by the training, validation and test set by the Levenberg-Marquardt algorithm.

Figure 5.8 shows the plotted regression R values for the 5 node network. For all sets, we can see there is a slight correlation error between target and outputs. However, it does show, that the network has found a close relationship between targets and outputs.

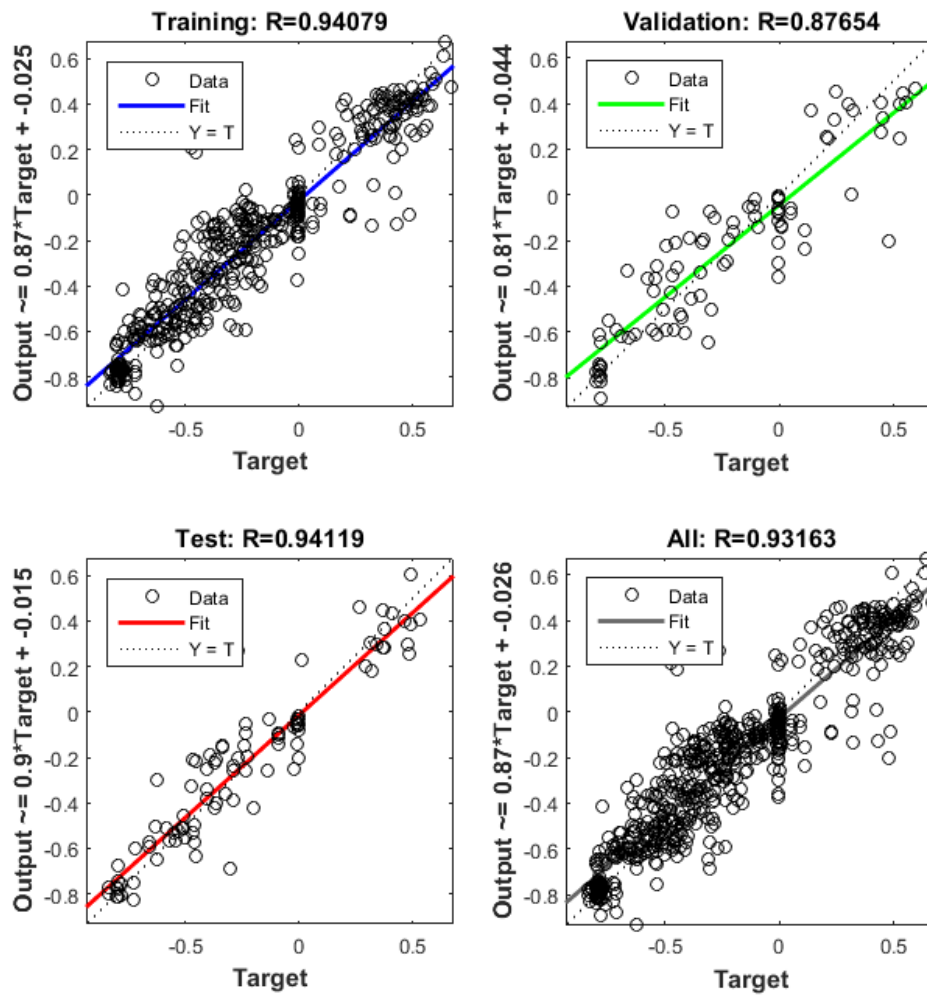


FIGURE 5.8: Regression plot of training, validation and test set from Matlab.

We want the network to generalize well. In the previous section we created 3 different networks with 3, 5 and 10 neurons in the hidden layer. If the number of neurons is too large, we might overfit the training data, and although we will minimize the MSE on training data, it will perform poorly on new data. To ensure the network has a good performance we will use an algorithm that can adjust the numbers of parameters in the network so that it generalizes well. The method we will use is called Bayesian regularization. See Section 4.5 for a description of the algorithm.

Training with Bayesian regularization

By using Bayesian regularization we get a larger training set because we do not need a validation set. The performance for different network sizes will now be compared as we did in the previous section when using the Bayesian regularization. The number of effective parameters γ from each network size will help us evaluate the size of the network. This parameter is described in Chapter 5.

We have divided the data into three sets as in the previous section. Although we do not need a validation set when using Bayesian regularization, we have included a small set with two samples to show that the algorithm gives 0 MSE and 1 for R:

- Training samples 80 % (34 samples).
- Validation samples 5 % (2 samples).
- Test samples 15 % (6 samples).

The mean square error, regression coefficient R and the effective number of parameters γ will be compared for 4 different networks with nodes equal to 3,5,10 and 20. Afterwards we will evaluate the different sizes and see which network is best.

	Samples [T]	MSE [-]	R [-]
Training	34	0.00694	0.978
Validation	2	0	0
Testing	6	0.0428	0.866

TABLE 5.4: Levenberg Marquardt backpropagation algorithm of network with 3 hidden neurons.

Where the effective number of parameters $\gamma = 96.1$ of total 115.

	Samples [T]	MSE [-]	R [-]
Training	34	0.00464	0.985
Validation	2	0	0
Testing	6	0.0165	0.949

TABLE 5.5: Levenberg Marquardt backpropagation algorithm of network with 5 hidden neurons.

Where the effective number of parameters $\gamma = 147$ of total 181.

	Samples [T]	MSE [-]	R [-]
Training	34	0.000993	0.996
Validation	2	0	0
Testing	6	0.0974	0.712

TABLE 5.6: Levenberg Marquardt backpropagation algorithm of network with 10 hidden neurons.

Where the effective number of parameters $\gamma = 282$ of total 346.

	Samples [T]	MSE [-]	R [-]
Training	30	4.55×10^{-17}	1.00
Validation	2	0	0
Testing	6	0.0904	0.744

TABLE 5.7: Levenberg Marquardt backpropagation algorithm of network with 20 hidden neurons.

Where the effective number of parameters $\gamma = 546$ of total 676.

Networks with 10 neurons or more overfit with a bad performance on the test data compared to the training data. We obtained the best result for a 5 neuron network. Using only 5 neurons reduces the time needed to train the network. For the 5 neuron network the MSE was 0.0165 and the correlation coefficient was close to 1 being $R = 0.949$ for the test samples. In Figure 5.9 we see a high correlation value for all the training samples for the 5 neuron network.

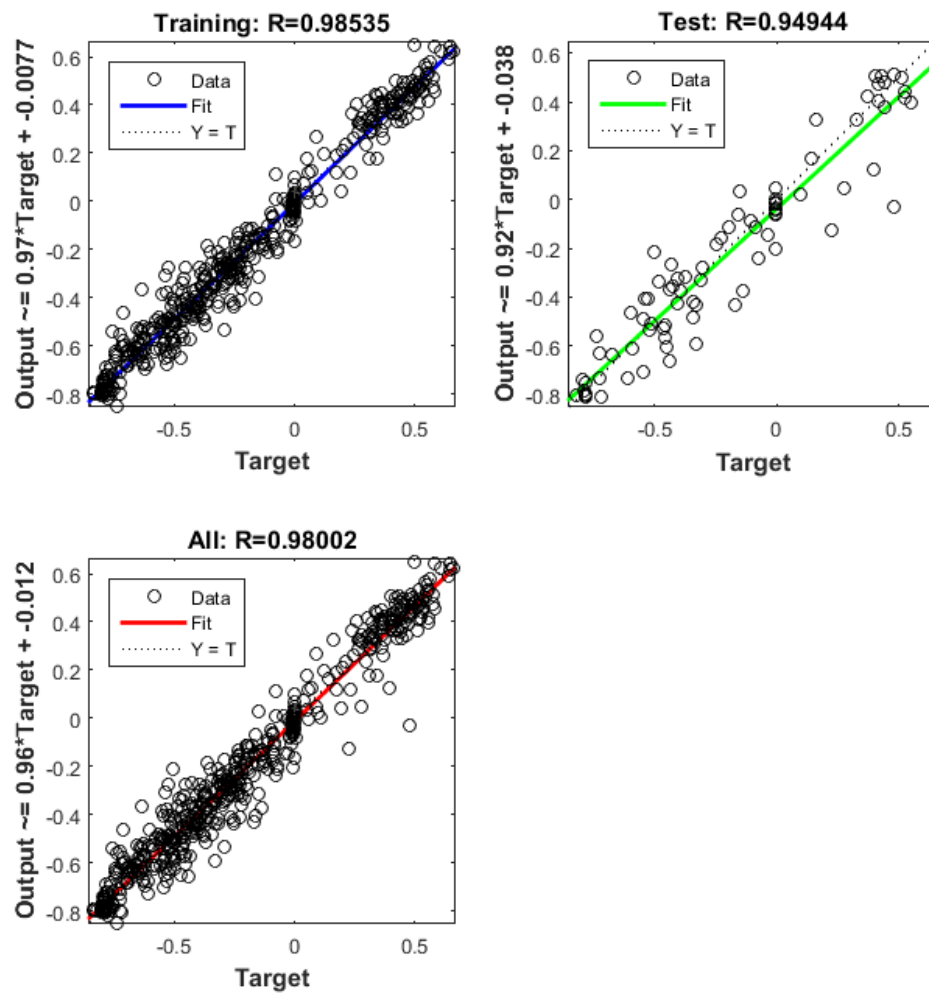


FIGURE 5.9: Regression plot of errors by the training, validation and test sets from Matlab.

Also in Figure 5.10 we see that the 5 neuron network has most of its errors close to zero. The number of errors steadily decrease the further we are from zero.

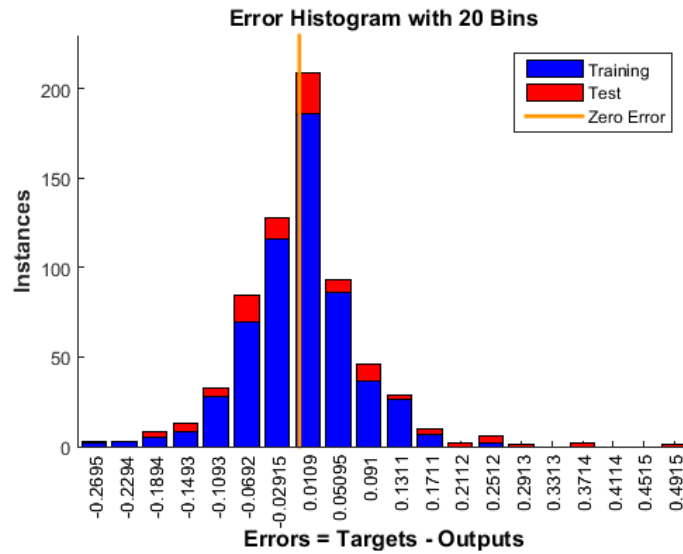


FIGURE 5.10: Histogram plot of errors by the training, validation and test sets from Matlab.

In Figure 5.11 we see that both test and training data give steady results after about 20 epochs.

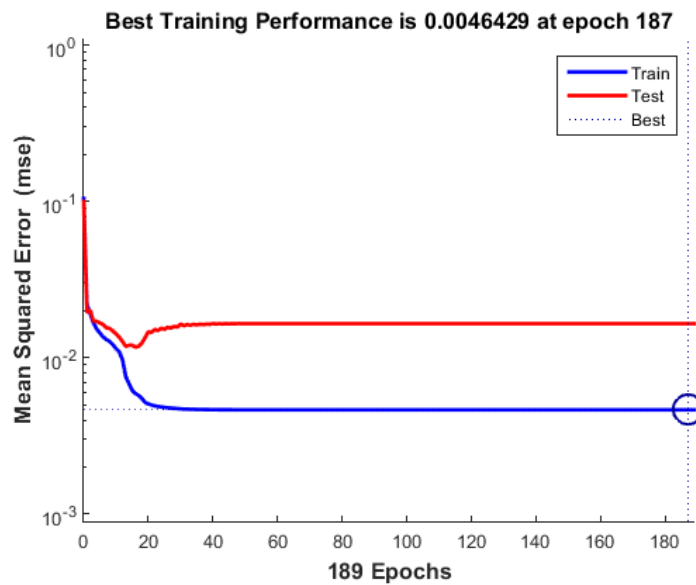


FIGURE 5.11: Performance plot of errors by the training, validation and test sets from Matlab.

5.4 Geometric shape results

Now we will present iceberg keel shapes created by the network and compare these results to the real geometric keel shape. We will use shape factors and area to evaluate the estimated iceberg keel shape. We will also train the same network several times, and use the average of the estimated keel shapes to compare with the real keel geometry.

Shape factors

Shape factors are dimensionless quantities we can use to describe the shape of an object, independent of its size. We will calculate the shape factors from the iceberg dimensions, such as diameter, chord lengths, etc. Here we will give a brief description of the shape factors used.

Aspect Ratio

The aspect ratio denotes the ratio of the major axis squared to the area. The major axis is the keel's greatest diameter. The formula for aspect ratio is given in (5.9):

$$\text{Aspect ratio} = \frac{\text{Major axis}^2}{\text{Area}} \quad (5.9)$$

Circularity

Circularity C , is defined as the degree to which the object is similar to a circle. Circularity is a measurement of both the particle form and roughness. The further away from a perfectly round and smooth circle the object becomes, the lower is the circularity value. In (5.10) a definition is given:

$$\text{Circularity} = \sqrt{\frac{4\pi \times \text{Area}}{P^2}} \quad (5.10)$$

The perimeter of the object, P , is defined as the total length of the objects boundary.

Convexity

The convexity shape factors rate how convex the geometry is by dividing the convex perimeter of the object with the real perimeter. A convex hull of an object can be thought of as a rubber band that surrounds the object.

$$Convexity = \frac{P_{convex}}{P} \quad (5.11)$$

In (5.11) P_{convex} represents the convex perimeter, and P represents the original perimeter of the object.

Solidity

We define solidity as the object area divided by the convex hull area as given in (5.12).

This shape factor is also dependent on the convex hull like convexity.

$$Solidity = \frac{Area}{Area_c} \quad (5.12)$$

$Area_c$ is the convex hull area.

Results

We have tested the network by training a 5 node network and removing 4 iceberg profiles (10 % of the total samples) to use for shape testing.

The original iceberg 2D cross-sections were normalized and reduced before applying them to the neural network. When comparing the results, we will therefore call the original iceberg keel parameters for **Real shape** and the reduced keel shape for **Reduced shape**. The estimated keel shape will be called **Estimated Shape**.

Below in the different tables we will compare the shape factors **Aspect ratio**, **Solidity**, **Circularity**, **Convexity**, and **Area**, with **Real Shape**, **Reduced Shape** and **Estimated Shape**.

Shape 1

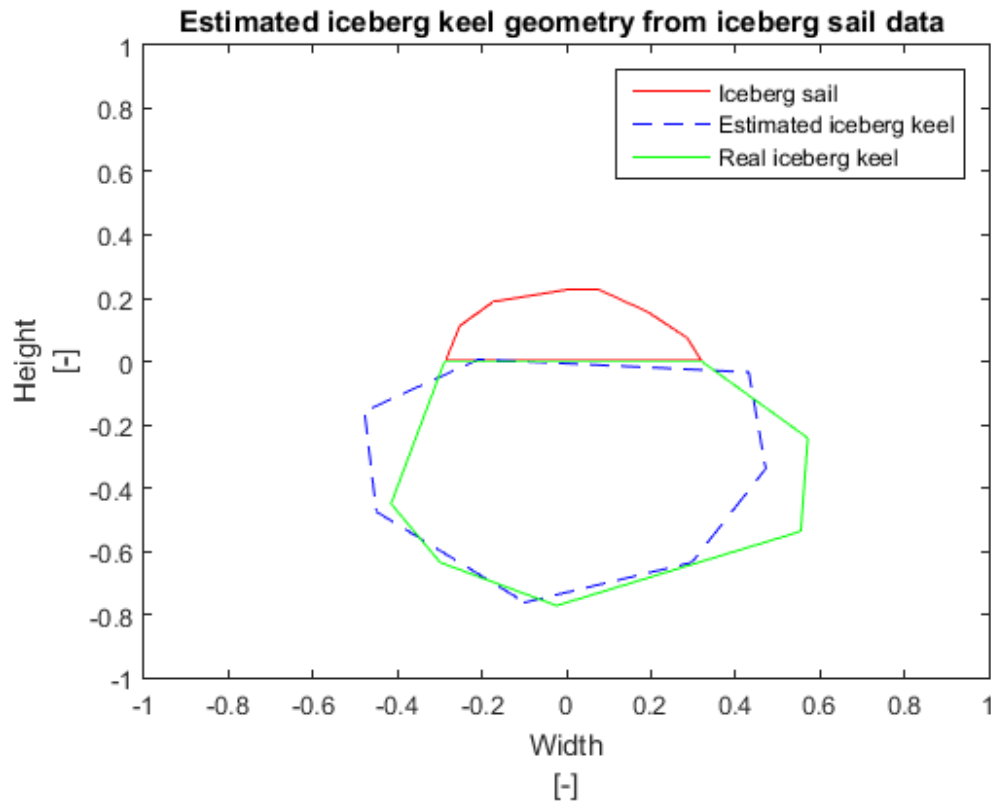


FIGURE 5.12: 2D cross sectional iceberg shape estimate from the Artificial Neural Network.

$$\% \text{ Difference} = \frac{\text{Estimated value} - \text{Real value}}{\text{Real value}} \times 100 \quad (5.13)$$

	Area	Aspect ratio	Solidity	Circularity	Convexity
Simplified Real	0.5822	1.7463	1.0000	0.9426	1.0000
Estimated	0.5516	1.7625	1.0000	0.9412	1.0000
% Difference	-5.2637 %	0.9316 %	0%	-0.1550 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real	$41.0445 * 10^4$	2.6341	0.9772	0.8392	0.9492
Estimated	0.5516	1.7625	1.0000	0.9412	1.0000
% Difference		-32.99 %	3.35 %	12.15 %	5.35 %

We can not compare the difference between real and estimated area because the estimated area has been normalized.

Shape 2

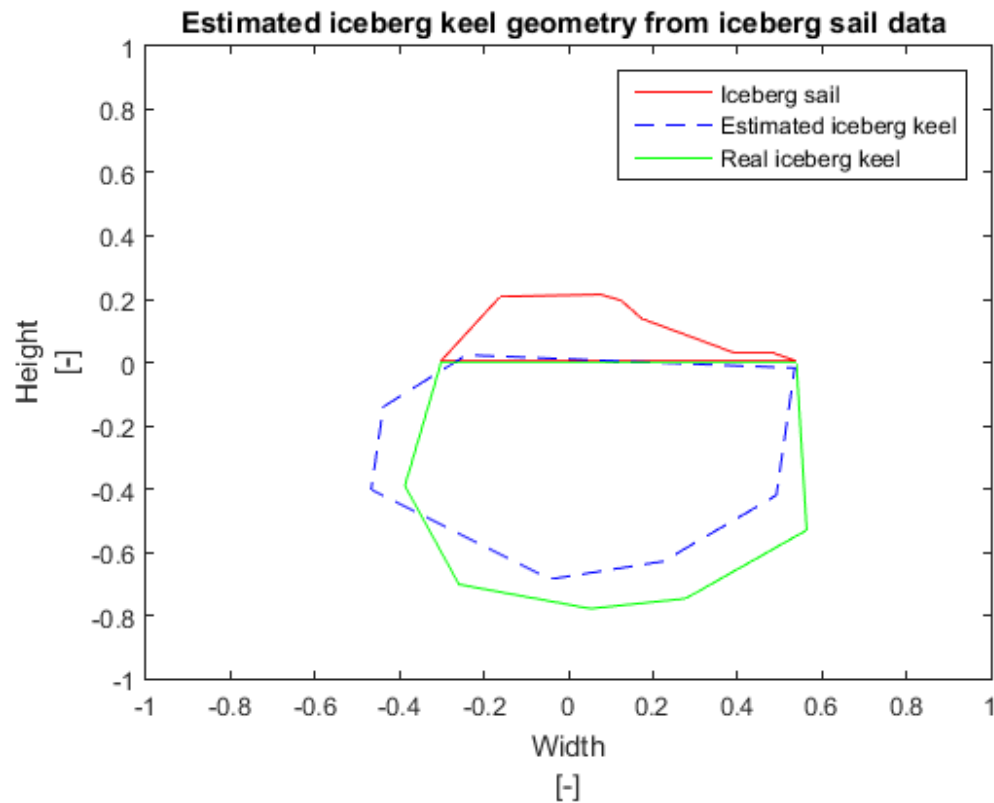


FIGURE 5.13: 2D cross sectional iceberg shape estimate from the Artificial Neural Network.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.6307	1.7899	1.0000	0.9343	1.0000
Estimated Shape	0.5516	2.1366	1.0000	0.9215	1.0000
% Difference	-14.4516 %	19.3722 %	0 %	-1.3774 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$47.0221 * 10^4$	1.6954	0.9810	0.9026	0.9575
Estimated Shape	0.5516	2.1366	1.0000	0.9215	1.0000
% Difference		26.02 %	1.93 %	-6.06 %	4.43 %

Shape 3

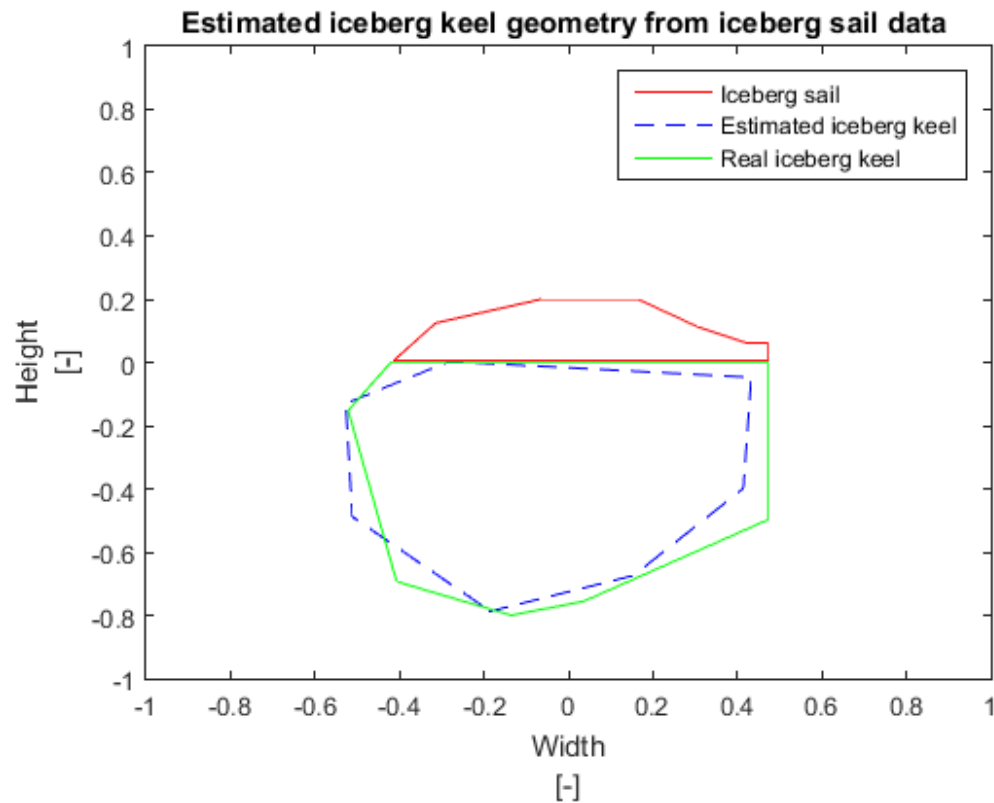


FIGURE 5.14: 2D cross sectional iceberg shape estimate from the Artificial Neural Network.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.6487	1.9277	1.0000	0.9214	1.0000
Estimated Shape	0.5732	1.8948	1.0000	0.9303	1.0000
% Difference	-11.6398 %	-1.7021 %	0%	0.9580 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.6332 * 10^4$	1.8835	0.9677	0.8918	0.9601
Estimated Shape	0.5732	1.8948	1.0000	0.9303	1.0000
% Difference		0.59 %	3.33%	4.31 %	4.15 %

Shape 4

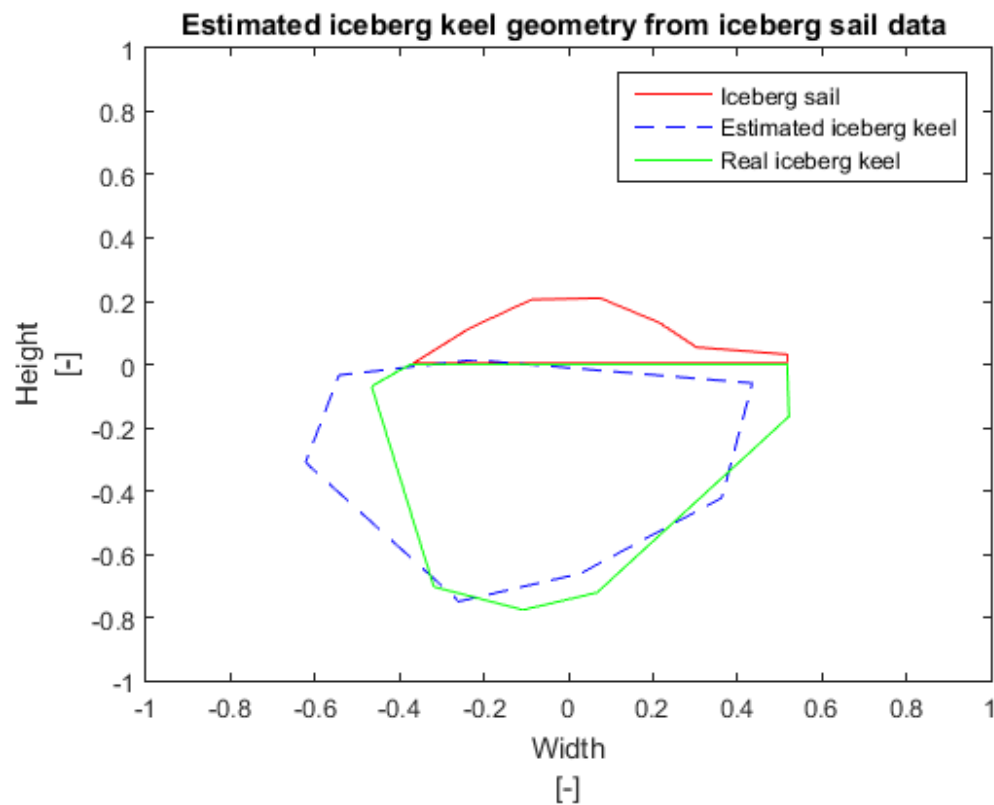


FIGURE 5.15: 2D cross sectional iceberg shape estimate from the Artificial Neural Network.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.5421	2.2036	1.0000	0.8851	1.0000
Estimated Shape	0.5626	2.1014	1.0000	0.9086	1.0000
% Difference	3.7861 %	-4.6378 %	0%	2.6602 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.7370 * 10^4$	2.1953	0.9696	0.8354	0.9353
Estimated Shape	0.5626	2.1014	1.0000	0.9086	1.0000
% Difference		-4.27 %	3.13 %	8.76 %	6.91 %

5.5 Discussion and improvements

The overall goal was to see if a neural network could estimate the shape of an iceberg keel, and also see how well the network performed. The idea was that there exists a relation between iceberg sail geometry and keel geometry. The arguments we use to justify such a relation are:

- We know there exist a relation to how much volume exist beneath the surface from assuming a uniform density of ice.
- By assuming the iceberg is in equilibrium. The center of buoyancy and center of gravity must ensure stability, which gives information about the distribution of volume.

The distribution of volume is very hard to model, because iceberg equilibrium stability can occur with many different distributions of volume. What we have done is to create an artificial network that estimates the most likely distribution, based on previous relations. As mentioned in the introduction, neural networks are useful when the underlying problem is hard to model, which irregular shaped icebergs are.

We estimated keel shapes from 4 different iceberg sails. When looking at Table 5.4 to Table 5.4 the network has an error of less than 15 % for most shape factors and areas when comparing the reduced shape with the estimated shape. When we compare the estimated shape with the real shape, the error is larger. This is because we lose information when we reduce the geometry. For the different estimated shapes we see that the shape factor for convexity and solidity are 1 or close to 1 for most cases. This is because we reduced the number of vertices from the original iceberg shape. When we reduce the vertices by calculating the importance of each vertex based on angle and segment length and then removing the least important one, we end up with more convex shapes.

We can achieve more accurate shape estimations by adding more parameters in the network. However, we would then need to increase the amount of neurons in the network, meaning we would need a larger data set to train the additional weights and biases.

One can argue that the sample set consisting of 42 2D iceberg geometries are very alike, and the shapes are not varied enough to give an actual representation of how irregular and diverse icebergs are in real life. This is true when we are only investigating 2D shapes compared to real life 3D geometries. It would be very interesting to see how the network will adapt to a larger data set with greater variation in geometry, and also the performance when we add a dimension to 3D.

Sensitivity towards stability

We will never achieve perfect results by estimation. However, it is interesting to see if the estimates are good enough for trajectory and stability calculations. In Figure 5.16 we have plotted the real geometry, the estimated geometry with a neural network, and the estimated trapezoidal geometry as was done in Chapter 3.

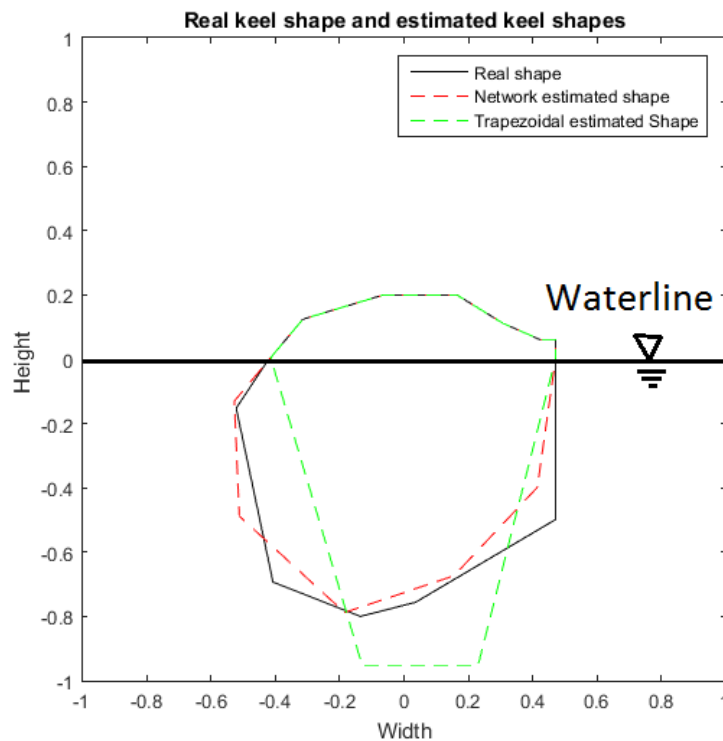


FIGURE 5.16: This figure compares the real geometry in black with estimates using different methods. Green is from estimating the keel to be a trapezoidal shape, and red if from using a neural network.

We have compared the GZ curves from the different shapes in Figure 5.17. In Figure 5.17 we see that the GZ curve from the neural network shape estimate fits better than the GZ curve from the trapezoid estimate. It should again be noted that the shapes and GZ values have been normalized and rescaled to make calculations easier. Also the trapezoidal estimate is not based on the same statistic as in Chapter 4, we have just used a factor that gives a fit for comparison.

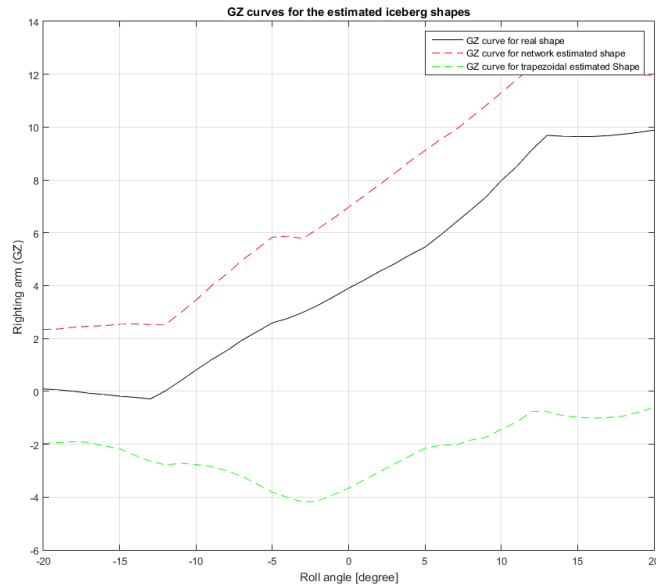


FIGURE 5.17: This figure compares the GZ curves from the real geometry in black, the trapezoidal shape estimate in green, and red from using a neural network.

3D analysis

The reason for performing the case study in 2D is because we do not have enough 3D iceberg data. In this section we will show how the neural network model can be generalized to a 3D iceberg model.

The challenge with creating a neural network to estimate a 3D keel is that the number of parameters will increase by adding an extra degree of freedom. In the previous case study we used (x,y) coordinates as input and target parameters for the network. Here we will use 3 (x,y,z) parameters for input and target -matrices.

By increasing the number of parameters in the network, the network would have to become larger, meaning more neurons. To train such a large network, we will

need a large data set. From the [Hodgson \[1988\]](#) data we have 3 samples of 3D icebergs, this is not enough data to train a network. However, we can show how we would prepare the data.

In [Figure 5.18](#) we can see a 3D model from [Hodgson \[1988\]](#). This model is created from 2047 (x,y,z) vertices. We create the geometry from a mesh by using Delaunay triangulation.

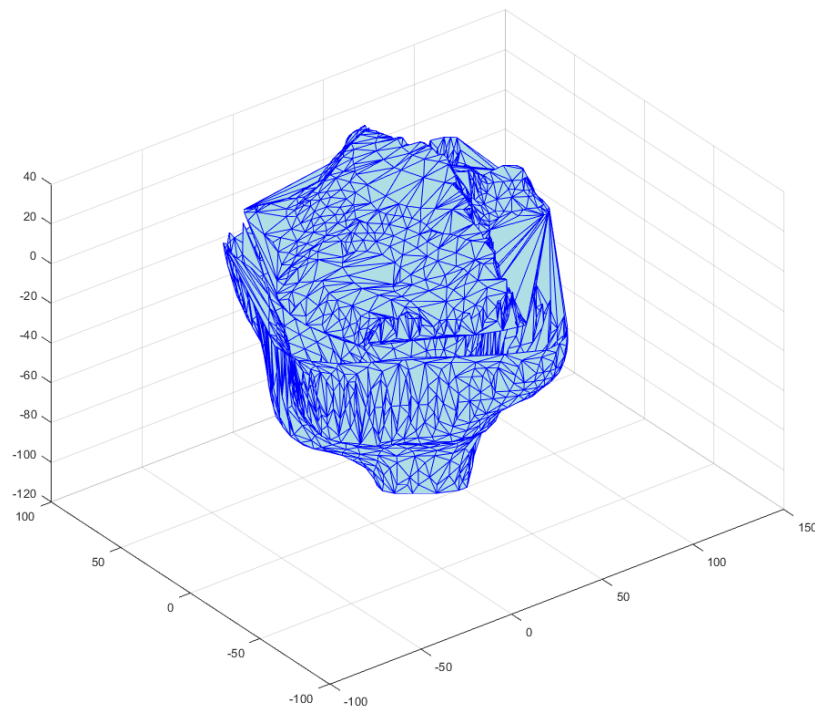


FIGURE 5.18: This model is created from 2047 (x,y,z) vertices using Delaunay triangulation.

The surface shown in [Figure 5.19](#) is of the iceberg keel. This geometry is created by a 904×3 matrix, containing 904 (x,y,z) vertices.

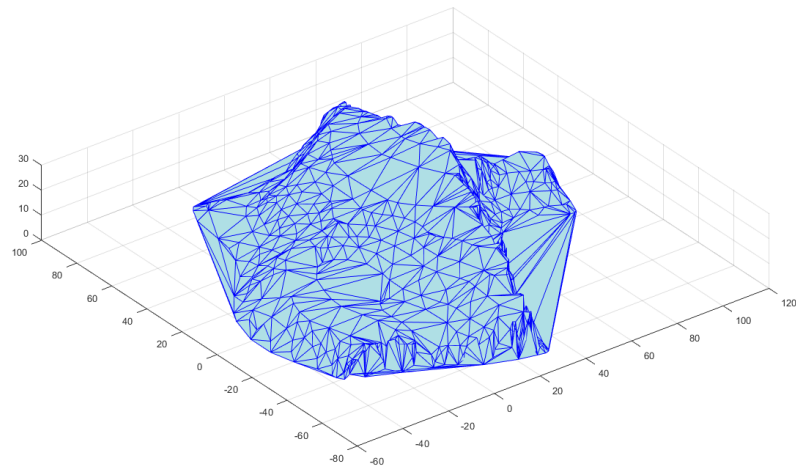


FIGURE 5.19: This iceberg sail geometry consist of 1756 faces and 904 vertices.

We must reduce the number of vertices in Figure 5.19 to limit the size of the neural network and number of parameters to estimate. We do this by using Delaunay triangulation in Matlab. In Figure 5.20 we can see the reduced geometry. It consists of 17 vertices describing the geometry. This means we will have a $[1 \times 51]$ ($17 \times x, y, xcoordinates$) input matrix for each iceberg sail.

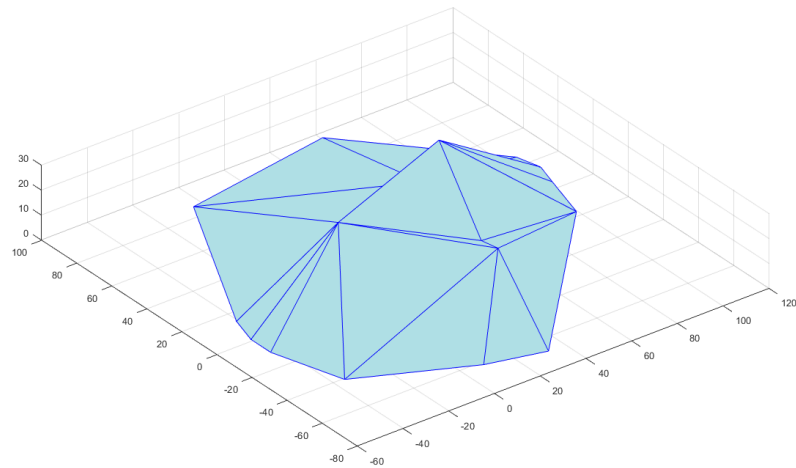


FIGURE 5.20: The new reduced iceberg geometry sail consist of 17 vertices and 20 faces.

We do the reduction for the iceberg keel, creating the Target 51 parameters ($17 \times x, y, z$ coordinates) large target matrix from 17 vertices. In Figure 5.21 we have the original iceberg keel with 1143 vertices and in Figure 5.22 we have the reduced iceberg keel geometry with only 17 vertices instead of using 904 vertices.

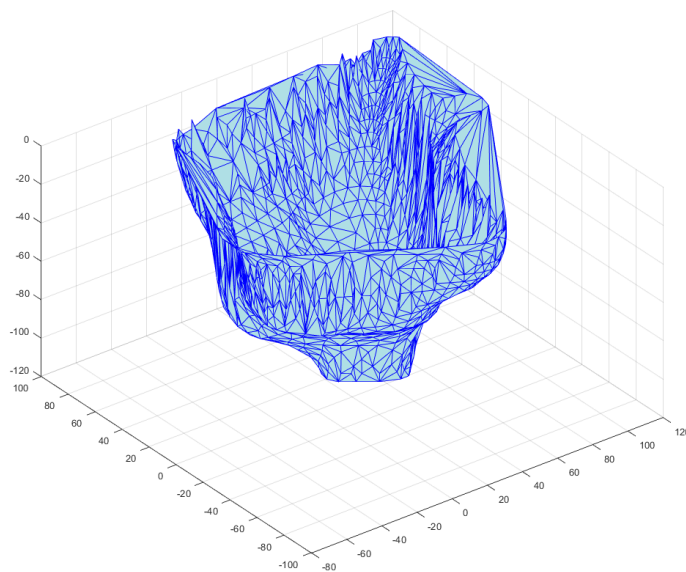


FIGURE 5.21: Original iceberg keel with 1143 vertices and 2248 faces.

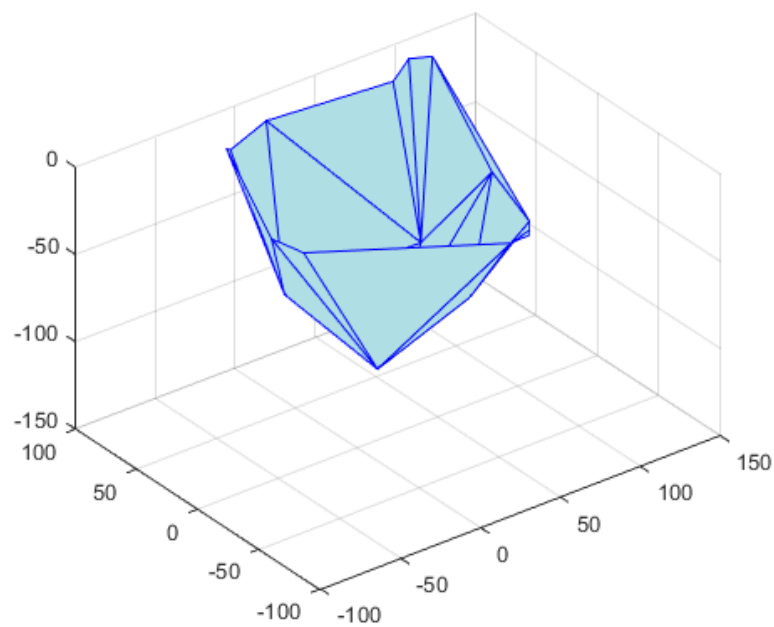


FIGURE 5.22: Reduced iceberg geometry for creating targets for the neural network. The geometry consist of only 17 vertices and 20 faces.

Create network

The next step would be to create a neural network for estimation. This is done by using the $[51 \times 1]$ input matrix representing vertices of the sail geometry with a $[51 \times 1]$ target matrix of vertices representing the keel geometry. The network can be trained by using the same backpropagation algorithm we used earlier, the Levenberg-Marquardt algorithm. The number of vertices were found through testing. We chose the least amount of vertices which still conserved most of the original geometry. The number of vertices we then chose to use was 17.

Accuracy

We will see how accurate the prepared data is compared to the original iceberg data. When using aerial stereo photographs the accuracy will be within $\pm 2\%$, while for the underwater water profiles the uncertainty is of about $\pm 15\%$ according to a report by [Hodgson \[1988\]](#).

In Table 5.5 the measurement accuracy is compared with the accuracy loss when reducing the number of vertices. The greatest total error is for the iceberg keel due to the poor accuracy of underwater scans.

	Original Sail	Reduced Sail	Original Keel	Reduced keel
Volume	$1.2213 \times 10^6 m^3$	1.1332×10^6	7.6350×10^5	7.4736×10^5
Accuracy from measurements	$\pm 2\%$		$\pm 15\%$	
Accuracy from reduction	-8.31 %		2.15 %	
Total accuracy	-10.31 %		17.15 %	

Improvements done to the neural network

Sample size

The neural network will perform better if we have a larger data set to train the network with. In this case we only had 42 samples for training, validation, and testing.

Network size

With a larger data set it would be interesting to see how we could optimize the network size to give even better estimations.

Train again

The network can always be trained again and again. We noticed when building the network that the mean square error varied each time we built, and retrained the network.

Add constraints

In all the estimated geometries there is some error in the surface where the keel does not match the surface geometry. This error can be removed by adding a constraint to the network.

Chapter 6

Case study: Online Estimation of Keel Geometry from an AUV Communication Link

Problem statement: An AUV can gather high-resolution data such as high-resolution sonar images and high-resolution snapshot camera images [Yoshida et al., 2011]. We often want to transmit data to a surface vessel in real time during a marine operation when the information is needed. However, the underwater communication is limited. Especially in the Arctic due to a multipath formation by the sound wave's repeated interactions with the ice cover [O'Hara and Collis, 2011]. If the AUVs uses an acoustic link, it may only transmit up to 80 kbps data at a limited range of 500 meters [Yoshida et al., 2011]. We want to simplify the data gathered by the AUV so it can be transmitted real time to the vessel for processing.

Proposed solution: The plan is to make the AUV calculate shape factors of the iceberg keel to transmit back to the surface vessel. We can then use the shape data with surface data taken from a UAV to predict the entire iceberg shape. This data can then assist us in towing operations and in calculating the iceberg's trajectory. We want to see how well a neural network can reconstruct keel shapes from iceberg keel shape factors.

In this case study 2D cross-sections are used for estimation. Later we will discuss how the method could be applied to a 3D case.

6.1 Preprocessing

We have used the same shape factors as in Chapter 5. The input matrix is 5 shape factors of 42 different iceberg keel, where the target is a matrix with 16 x y -coordinates of 42 iceberg keels.

Input matrix:

$$P = [5 \times 42]. \quad (6.1)$$

Target matrix:

$$P = [16 \times 42]. \quad (6.2)$$

We have divided the image data into three sample sets:

- Training samples 80 % (34 samples).
- Validation samples 5 % (2 samples).
- Test samples 15 % (6 samples).

6.2 The network

We are using the same design on the network as in Chapter 6. We will have the same number of layers and neurons.

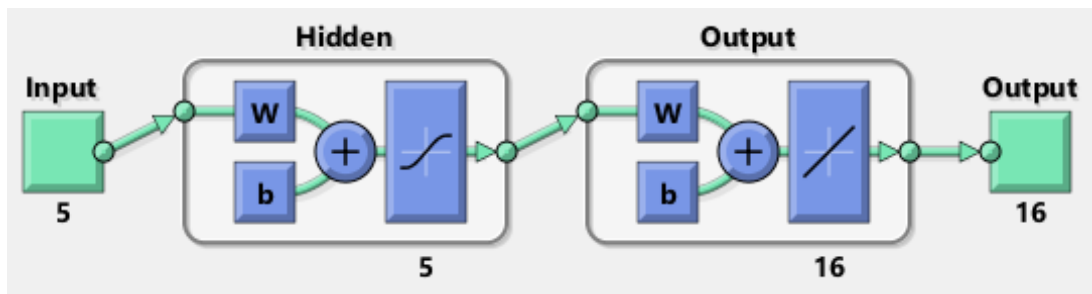


FIGURE 6.1: Conceptual design drawing of network, with 5 neurons in hidden layer.

Training

10 node network:

Where the effective number of parameters $\gamma = 282$ of total 346.

	Samples [T]	MSE [-]	R [-]
Training	34	0.00468	0.985
Validation	2	0	0
Testing	6	0.068	0.7869

TABLE 6.1: Levenberg Marquardt backpropagation algorithm with Bayesian regularization of network with 10 hidden neurons.

5 node network:

	Samples [T]	MSE [-]	R [-]
Training	34	0.0106	0.966
Validation	2	0	0
Testing	6	0.0233	0.924

TABLE 6.2: Levenberg Marquardt backpropagation algorithm with Bayesian regularization of network with 5 hidden neurons.

3 node network:

	Samples [T]	MSE [-]	R [-]
Training	34	0.0127	0.959
Validation	2	0	0
Testing	6	0.0535	0.825

TABLE 6.3: Levenberg Marquardt backpropagation algorithm with Bayesian regularization of network with 3 hidden neurons.

A 5 node network gave the best results, since it performed well on the test data, with a high correlation coefficient R.

Here we will give some performance plots.

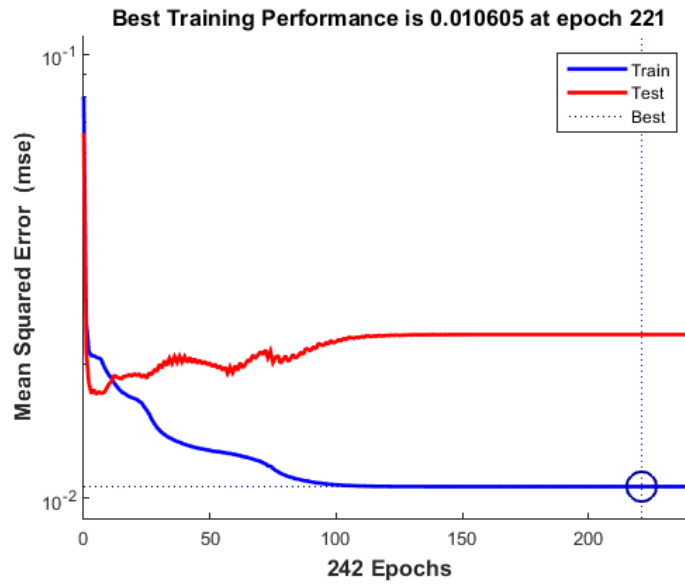


FIGURE 6.2: Performance plot of the 5 node network from Matlab.

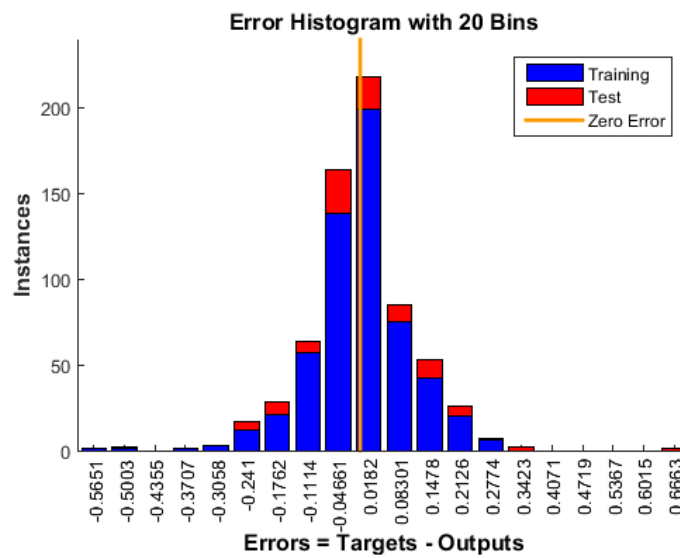


FIGURE 6.3: Histogram plot of the 5 node network from Matlab.

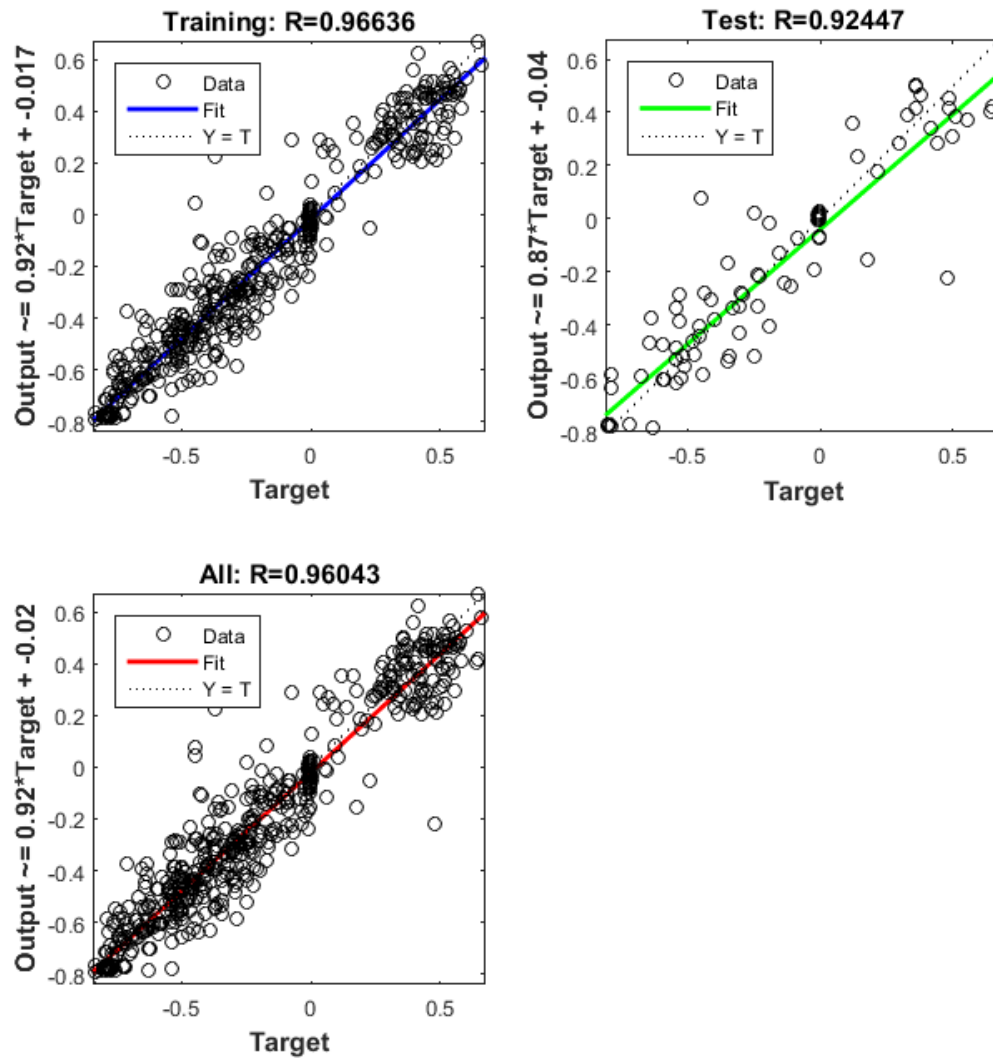


FIGURE 6.4: Regression plot of the 5 node network from Matlab.

Results

We have tested the network by training a 5 node network and removing 4 iceberg profiles (10 % of the total samples) to use for shape testing.

The 2d cross-sections have been normalized and reduced before applying them to the neural network. Therefore when comparing the results, we will call the original iceberg keel parameters for **Real shape** and the reduced keel shape for **Reduced shape**. The estimated keel shape will be called **Estimated Shape**.

Below in the different tables we will compare the shape factors **Aspect ratio**, **Solidity**, **Circularity**, **Convexity**, and **Area**, with **Real Shape**, **Reduced Shape** and **Estimated Shape**. The equations can be found in Chapter 5.

The calculated difference in each table is calculated with (5.13).

Shape 1

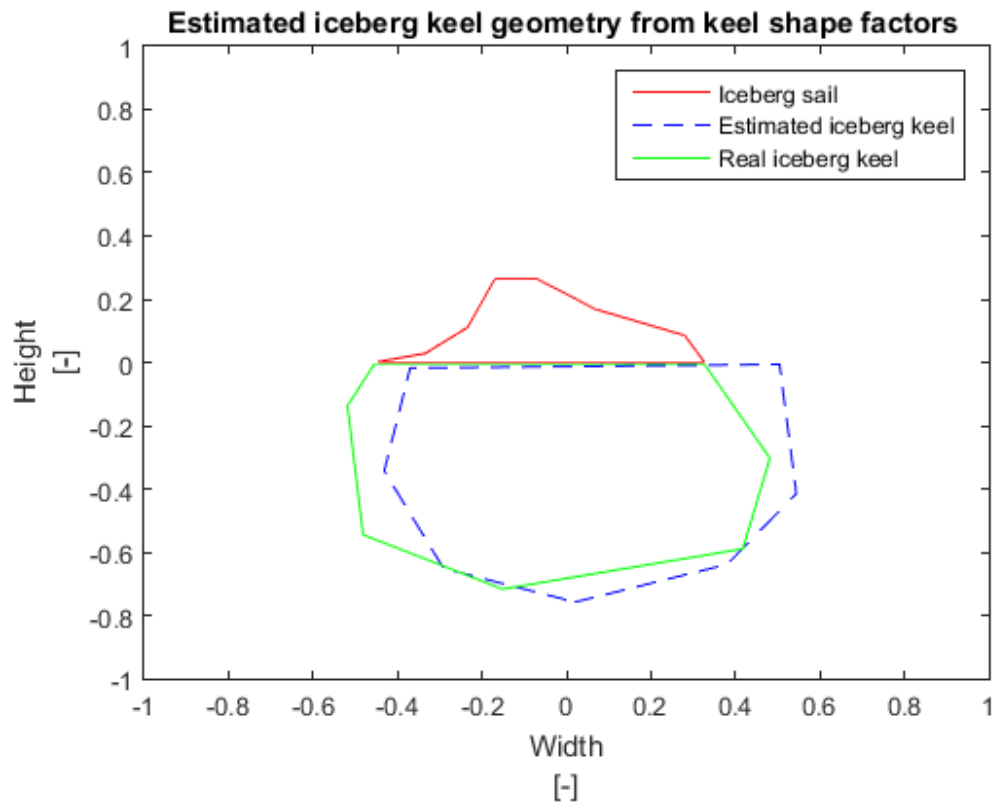


FIGURE 6.5: Keel shape estimate from shape factors and area data. Plot from Matlab.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.5899	1.8697	1	0.9312	1
Estimated Shape	0.5974	1.7568	1	0.9314	1
% Difference	1.2691 %	1.7568 %	0%	0.0293 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.1884 * 10^4$	1.6349	0.9653	0.8892	0.9346
Estimated Shape	0.5974	1.7568	1	0.9314	1
% Difference		7.45 %	3.59 %	4.74 %	6.99 %

Shape 2

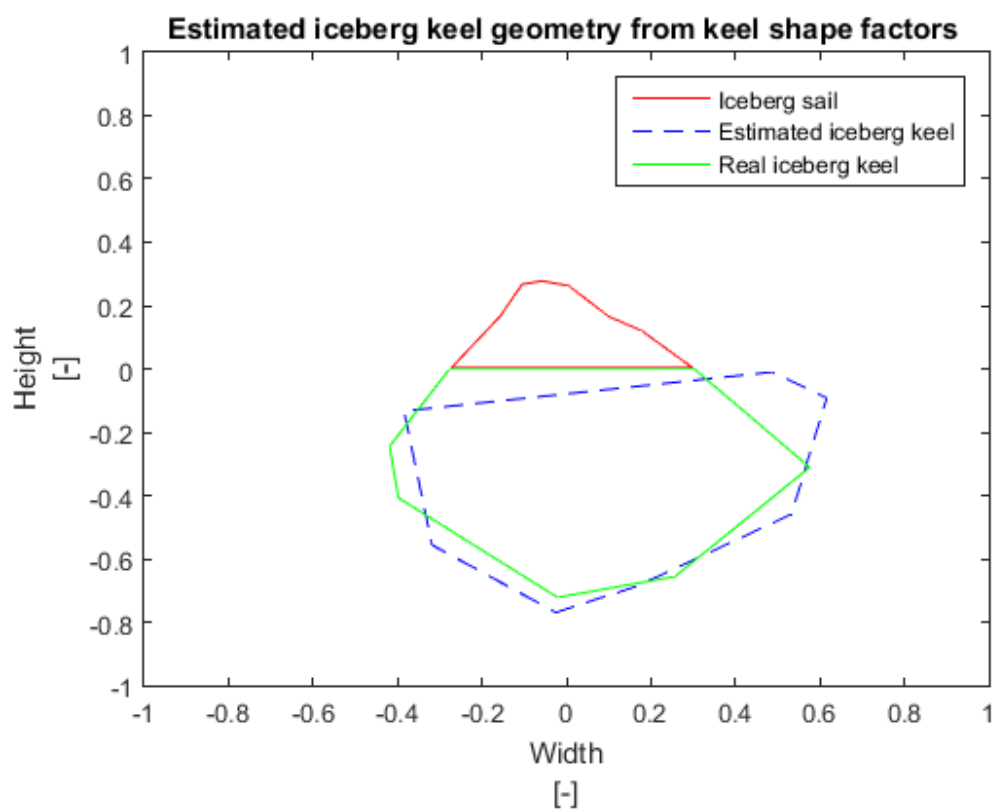


FIGURE 6.6: Keel shape estimate from shape factors and area data. Plot from Matlab.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.5047	1.9693	1	0.9378	1
Estimated Shape	0.5205	2.0923	1.0000	0.9004	1.0000
% Difference	3.1211 %	6.2457 %	0%	-3.98563 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.1593 * 10^4$	1.8491	0.9468	0.8320	0.8918
Estimated Shape	0.5205	2.0923	1.0000	0.9004	1.0000
% Difference		13.15 %	5.66 %	8.22 %	12.13 %

Shape 3

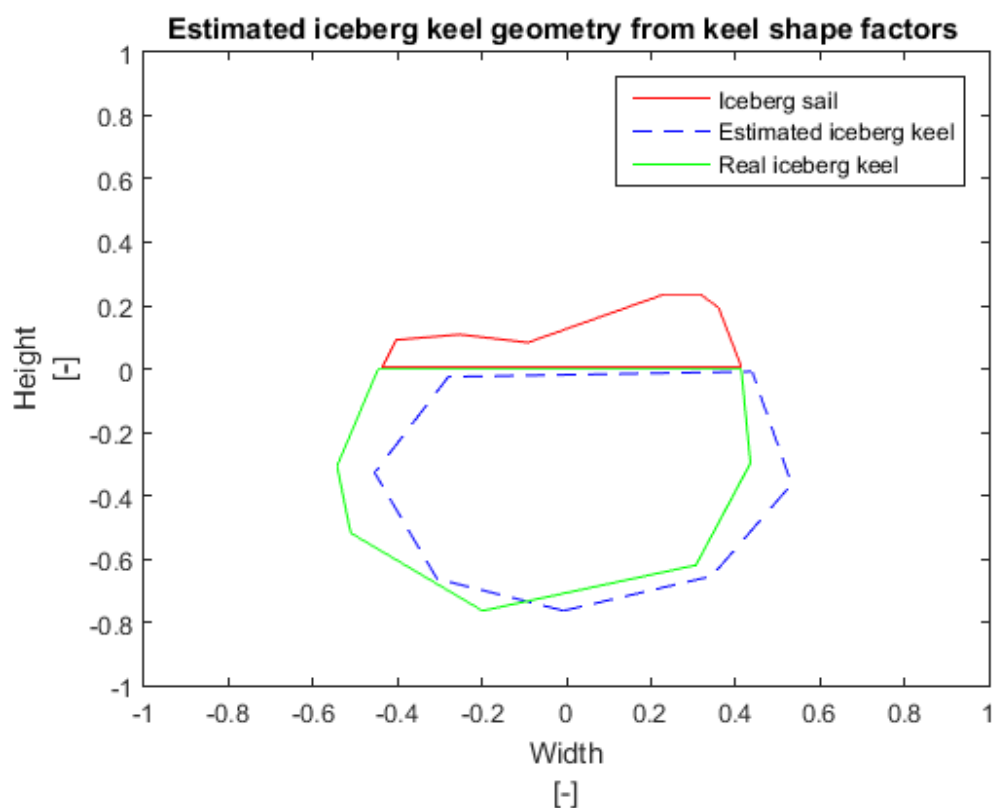


FIGURE 6.7: Keel shape estimate from shape factors and area data. Plot from Matlab.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.6029	1.8615	1	0.9295	1
Estimated Shape	0.5720	1.7229	1	0.9462	1.0000
% Difference	-5.1293 %	-7.4489 %	0%	1.789 %	0 %

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.5795 * 10^4$	1.7413	0.9655	0.8757	0.9356
Estimated Shape	0.5720	1.7229	1	0.9462	1.0000
% Difference		-1.05 %	3.57 %	8.05 %	6.88 %

Shape 4

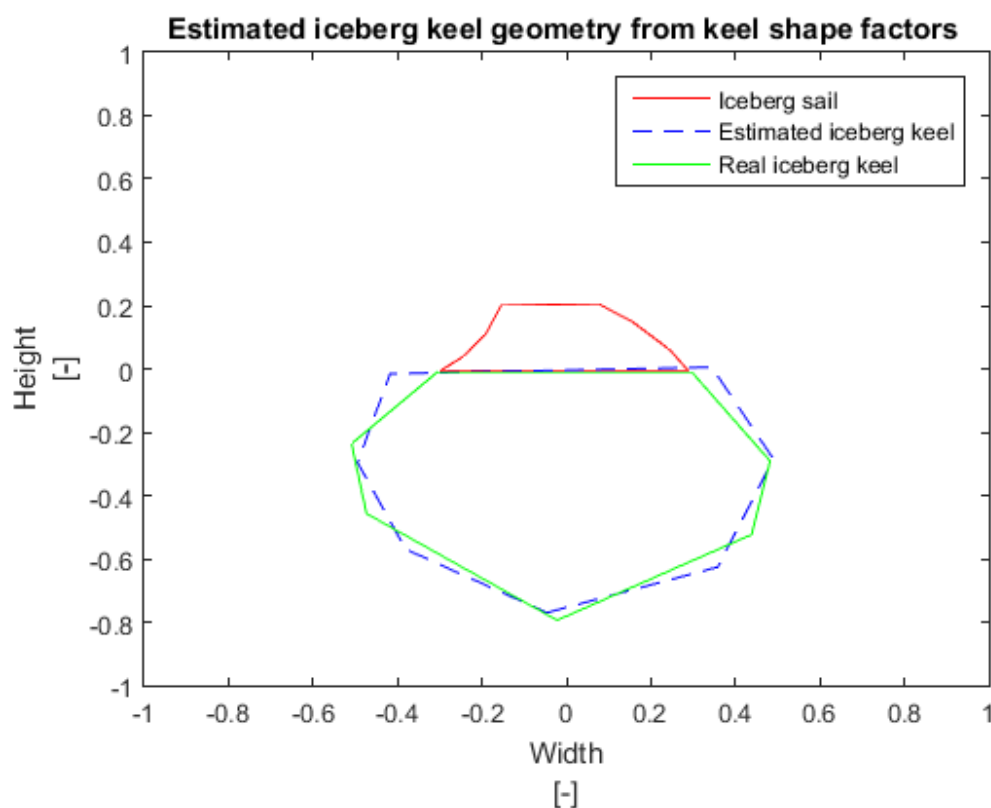


FIGURE 6.8: Keel shape estimate from shape factors and area data. Plot from Matlab.

	Area	Aspect ratio	Solidity	Circularity	Convexity
Reduced Shape	0.5571	1.7690	1	0.9446	1
Estimated Shape	0.5795	1.6820	1	0.9438	1.0000
% Difference	4.0219 %	-4.9187 %	0%	-0.0811 %	0%

	Area	Aspect ratio	Solidity	Circularity	Convexity
Real Shape	$41.2893 * 10^4$	1.9719	0.9566	0.8622	0.9311
Estimated Shape	0.5795	1.6820	1	0.9438	1.0000
% Difference		-14.70 %	4.53%	9.46 %	7.39 %

6.3 Discussion and improvements

The overall goal was to see if a neural network could recreate a keel geometry from 5 parameters using a neural network. The network was able to do this by using shape factors and the area as input parameters. The error was within $\pm 10\%$ when comparing the results to the real shape factors and areas. However, when looking at the plot in Figure 6.6, the volume distribution does not coincide close to the surface. This would give bad stability estimates, since the volume distribution close to the surface affects the initial stability a lot.

3D Analysis and improvements on the Neural Network

The shape factors used, in this case, can also be made for 3D geometries. The challenge will be the same as for the case in Chapter 5. We will have more parameters to estimate in the 3D case, this can create bad estimation if we do not have a large data set to train the network

The improvements to the neural network would be the same as for the network in Chapter 5.

6.4 Conclusion

Although a neural network can be used to reduce the geometry for transmitting the geometry data underwater with some error, we believe this is not the way to go. In a marine operation it is mainly trajectory calculations and stability of an iceberg for towing we are interested in. Therefore it would be easier to have a small computer onboard the AUV to directly calculate iceberg stability and trajectory calculations with a high resolution keel geometry with additional local

measurements, for example current measurements. The AUV would then only have to transmit a couple of parameters describing the onboard calculated stability and trajectory of the iceberg. In addition, if there is valuable data gathered from other sensor platforms this data could be added together with the AUV data onboard the surface vessel.

Conclusion

In Chapter 3 we investigated the application of keel shape estimations for calculating thfor a towing operation. When comparing the real GZ curve to the estimated one, we had a large error. Using a predetermined geometry such as a trapezoid and just scaling the trapezoid based on surface length did not give good stability estimations. The method could be improved by using a neural network, as was demonstrated in Figure 5.16, when comparing the GZ curves for the two estimation methods.

The proposed solution to the case study in Chapter 5 worked by using a neural network to estimate the keel shape, and was considerably more accurate than the trapezoidal estimation method. However, many challenges arise when relying on an aerial sensor platform which is needed to create the DEM. For example, harsh weather conditions, landing and deploying an aerial vehicle, reduced range and GPS vulnerability are just some of the challenges relying on an aerial sensor platform for ice management.

In the final case study we investigated the use of a neural network to rebuild a keel geometry from shape factors. The advantage of this method is so that less information needs to be transmitted from the AUV to the surface vessel. Instead of transmitting high-resolution data, the AUV could send important shape factors instead. The network was able to reconstruct the geometry, but it was hard to evaluate the accuracy apart from looking at the results visually, since we already used shape factors for input parameters. During this case we realized that instead of transmitting the entire geometry or parts of it, the AUV can use the high resolution data together with current data and just send a vector back to the surface vessel describing the icebergs drift direction. Onboard the vessel other data can then be added, such as for example wind speed and surface geometry of the iceberg, if available.

Further work

To reach the goal of creating a cloud-based neural network where industry stakeholders and research institutions can share their iceberg data, we need to further explore the networks performance on 3D data. We also need to test the performance of other estimation methods to see if a neural network is the best tool for this specific problem.

In order to assist the AUV to transmit hydro acoustic data, the dynamic model can be calculated onboard the AUV, where we only transmit the results from the dynamic model. It would then be interesting to see how the results from this dynamic model can be fused with data from other sensors onboard the surface vessel.

Appendix A

Overview of objects and their shape factors from [Olson, 2011].

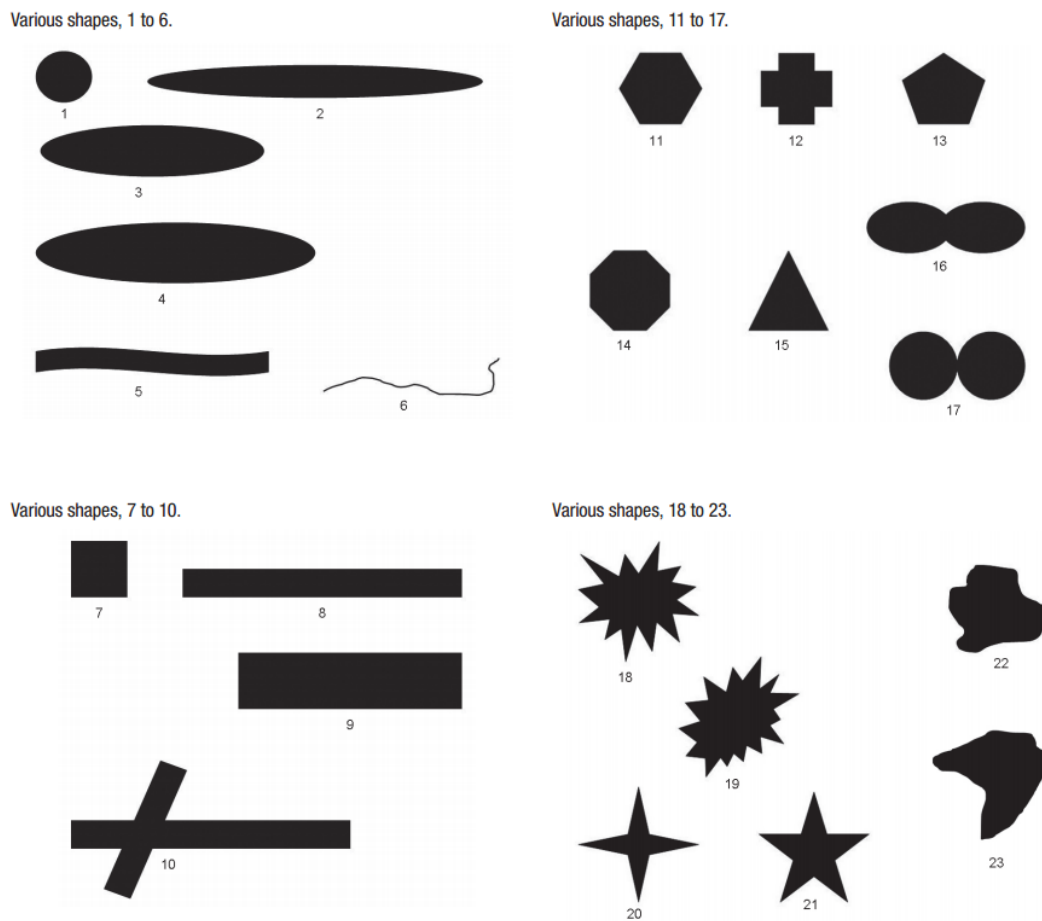


FIGURE A.1: Different objects with a number that indicates a shape factor in the table below.

ID	Shape Description	Aspect Ratio	Circularity	Convexity	Solidity
1	Circle	1	1	1	1
2	Torpedo	0.10	0.49	1	1
3	Long ellipse	0.22	0.69	1	1
4	Medium ellipse	0.23	0.71	1	1
5	Ribbon	0.11	0.49	0.99	0.85
6	Fiber	0.20	0.14	0.89	0.10
7	Square	1	0.89	1	1
8	Long rectangle	0.10	0.51	1	1
9	Medium rectangle	0.25	0.71	1	1
10	Twinning	0.49	0.45	0.82	0.47
11	Regular hexagon	0.90	0.95	1	1
12	Symmetric cross	1	0.79	0.87	0.87
13	Regular pentagon	0.86	0.93	1	1
14	Regular octagon	1	0.97	1	1
15	Equilateral triangle	1	0.78	1	1
16	Peanut	0.33	0.77	0.96	0.92
17	Touching circles	0.50	0.75	0.87	0.88
18	Star	1	0.42	0.58	0.59
19	Star	0.81	0.49	0.63	0.69
20	4-point star	0.96	0.45	0.83	0.38
21	5-point star	0.98	0.52	0.81	0.50
22	Random shape	0.94	0.86	0.94	0.91
23	Random shape	0.92	0.79	0.94	0.82

TABLE A.1: Table showing different shape factors for the objects shown in the previous figure.

Bibliography

- Garry Timco. *Compilation of iceberg shape and geometry data for the grand banks region*, volume PERD/CHC Report 20-43. National Research Council Canada, 1999.
- C-CORE. Ice management for structures in sea ice with ridges and icebergs. Volume 1 State of the Art in Iceberg Management(709), 2007.
- A. Palmer and K. Croasdale. *Arctic Offshore Engineering*. World Scientific, 2013. ISBN 9789814368773. URL <http://books.google.no/books?id=AQoCvss7dhEC>.
- P. Wadhams. *Ice in the Ocean*. Taylor & Francis, 2000. ISBN 9789056992965. URL <http://books.google.no/books?id=CQli-DiBqcQC>.
- Richard McKenna. Development of iceberg shape characterization for risk to grand banks installations. 2004.
- H Keys. Towards a new shape classification of antarctic icebergs. *Iceberg Res*, 12: 15–19, 1986.
- Garry Timco. Grand banks iceberg management. 2007.
- Garry Timco. Perd iceberg management database update through 2012 ice season. 2013.
- Peter Corke. *Robotics, vision and control: fundamental algorithms in MATLAB*, volume 73. Springer, 2011.
- Joakim Haugen. Autonomous Aerial Ice Observation. (November), 2014.
- Randal W Beard and Timothy W McLain. *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.

Gianpaolo Conte. *Vision-Based Localization and Guidance for Unmanned Aerial Vehicles* Gianpaolo Conte. Number 1260. 2009. ISBN 9789173936033.

Stuart D Smith, Norman R Donaldson, Chemical Sciences Branch, and Scotiafundy Region. Dynamic modelling of iceberg drift using current profiles. (91), 1987.

R.C Bailey. Implications of iceberg dynamics for iceberg stability estimation. *Cold Regions Science and Technology*, 22(2):197–203, 1994. doi: 10.1016/0165-232X(94)90028-0.

D.W. Bass and D.R. Attwood. Iceberg stability an error analysis. *Cold Regions Science and Technology*, 13(1):49–55, 1986.

Simon Haykin. *Neural Networks and*, volume 5. 2001. ISBN 9780131471399.

Artificial neural network with layer coloring.

M T Hagan, H B Demuth, and M H Beale. *Neural Network Design*. (. 10), 2002. URL <http://books.google.ru/books?id=bUNJAAAACAAJ>.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Marvin Minsky. S. papert, perceptrons, 1969.

DE Rumelhart GE Hinton RJ Williams and GE Hinton. Learning representations by back-propagating errors. *Nature*, pages 323–533, 1986.

David Parker. *Learning-logic: Casting the cortex of the human brain in silicon*. 1985.

Y Le Cun. Une procedure d'apprentissage pour reseau a seuil assymetrique. *Proceedings of Cognitiva*, 85:599–604, 1985.

David JC MacKay. Bayesian interpolation. *Neural computation*, 4(3):415–447, 1992.

Peter Bone. Polygon simplification. <http://www.mathworks.com/matlabcentral/fileexchange/45555-polygon-simplification>, 2014.

GJ Hodgson. *The dynamics of iceberg grounding and scouring (DIGS) experiment and repetitive mapping of the eastern Canadian continental shelf*. Number 94. Environmental Studies Research Funds, 1988.

Hiroshi Yoshida, Tadahiro Hyakudome, Shojiro Ishibashi, Hiroshi Ochi, Kenichi Asakawa, Takafumi Kasaya, Takashi Saito, and Shogo Okamoto. Study on land-to-underwater communication. *2011 The 14th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–5, 2011. ISSN 1347-6890.

C. a. O’Hara and J. M. Collis. Acoustics in Arctic Environments. *162nd Acoustical Society of America Meeting*, pages 1–2, 2011. URL [http://inside.mines.edu/~sim\\$jcollis/Pekerice_Lay_Language_Paper.pdf](http://inside.mines.edu/~sim$jcollis/Pekerice_Lay_Language_Paper.pdf).

Eric Olson. Particle Shape Factors and Their Use in Image Analysis Part 1 : Theory. *Journal of GXP Compliance*, 15(3):85–96, 2011. ISSN 1091-3483. URL <http://www.ivtnetwork.com/article/particle-shape-factors-and-their-use-image-analysispart-1-theory>.