NTNU

Norwegian University of
Science and Technology

Estimating the Design Parameters of a Highly Skewed Ship Propeller by Automated 3D-scanning

Njål Haagensli Munthe-Kaas

Subsea Technology
Submission date: June 2018
Supervisor: Olav Egeland, MTP

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Summary

Ship propellers have a highly complex surface geometry. Certain design parameters are commonly used to describe this complex geometry. During production of the propeller blades there is a need for measuring the blade surfaces to ensure that the blades are manufactured according to the standard tolerances. Current measurement techniques are slow and can only measure discrete points on the surface. Therefore, the propeller manufacturing industry is interested in finding measurement solutions that can measure the whole blade surface at once. 3D scanners have the potential of measuring the whole blade surface in a short amount of time. But one thing is to capture the geometry data itself, another is to be able to do something useful with this data.

This is where the design parameters of the propellers come in. If these parameters can be estimated from the data a 3D scanner captures, this could be used to measure the propeller blades during manufacturing. The blades could then be grinded down until the measured parameters are in accordance with the design data. Such a system could also be used for reverse engineering of propellers, with the goal of creating a CAD model of a propeller when no design drawing for the propeller is present.

In this report, an automatic scanning system consisting of a structured light 3D scanner, an industrial robot and a laser distance sensor is presented. A controllable pitch propeller blade is scanned using an algorithm for automatic generation of the robot scan poses. The challenges of scanning a controllable pitch propeller blade is described. Then, a method for estimating the design parameters of the propeller based on the scan is presented. The relevant theory describing how the parameters are defined is outlined.

Experimental results from the estimation of the design parameters are presented, and compared to results from a high-end optical metrology system.

The results produced by the developed system are encouraging, considering they are based on the first few tests. With some improvements the system could prove to be a much cheaper solution for in-line measurements and reverse engineering of propellers than comparable systems. The system also looks to be easy to integrate with a grinding robot, which means we could see a robot cell with completely automated grinding and finishing of propeller blades in the future.

Sammendrag

Skipspropeller har en svært kompleks overflategeometri. Såkalte designparametre blir ofte brukt for å beskrive denne komplekse geometrien. Under produksjon av propellblader er det behov for en oppmåling av bladene for å sørge for at de er produsert i henhold til de gjeldende toleransene. Nåværende måleteknikker er treige og kan bare måle ett diskret punkt på overflaten av gangen. Derfor er propellprodusentene interessert i å finne løsninger for oppmålingen hvor man kan måle hele bladoverflaten på en gang. 3D-scannere har potensiale til å kunne måle et helt propellblad på kort tid. Men en ting er å kunne fange geometridata av bladet, noe annet er å kunne gjøre noe nyttig med disse dataene.

Det er her designparametrene for propellen kommer inn. Hvis disse parametrene kan bli estimert fra dataen som en 3D-scanner fanger, kan dette brukes til å måle propellbladene under produksjonen. Bladene kan da bli slipt ned helt til de estimerte parametrene stemmer overens med dataene fra propelldesigneren. Et slikt system kan også brukes til reverse engineering av propeller, med mål om å lage en CAD-modell av en propell når man mangler geometridata eller en propelltegning.

I denne rapporten blir et automatisk scannesystem bestående av en 3D-scanner basert på strukturert lys, en industrirobot, og en laseravstandsmåler presentert. Et propellblad blir scannet ved bruk av en algoritme som automatisk genererer scanneposisjonene til roboten. Utfordringene ved å scanne et blad fra en propell med kontrollerbar pitch blir beskrevet. Så blir en metode for å estimere propellens designparametrene ut i fra 3D-scannen presentert. Teorien som beskriver hvordan parametrene er definert blir også beskrevet.

Eksperimentelle resultater fra estimeringen av designparametrene blir presentert, og sammenlignet med resultatene fra et optisk scannesystem som representerer state of the art innen feltet.

Resultatene er lovende med tanke på at de er basert på de aller første testene av systemet. Med noen forbedringer kan systemet vise seg å være en mye billigere løsning enn sammenlignbare systemer. Systemet ser også ut til å kunne settes sammen med en sliperobot. Det betyr at vi i fremtiden kanskje kan få se en robotcelle med fullstendig automatisert sliping og ferdiggjøring av propellblader.

Preface

This document is the thesis concluding my M.Sc. degree in Subsea Technology at the Norwegian University of Science and Technology (NTNU). The work presented was carried out between January and June 2018.

My motivation for choosing this research topic was based on an interesting research project with reverse engineering, fascination with the applications of the field of computer and robot vision, as well as a personal interest in boats, marine technology and the sea in general.

I would like to thank PhD candidate Eirik Njåstad for a great amount of help with developing the automatic scanning procedure, and useful discussions on how the different problems can be solved.

I also want to thank my supervisor for this project, Professor Olav Egeland, for facilitating the work in a good way and providing the necessary equipment for the project.

The industrial partner Oshaug Metall AS, with CEO Stein Berg Oshaug and the other pleasant employees, have been very helpful in providing the propeller blade for the scanning. They have also responded positively to any questions and requests made, and provided necessary information on the propeller blade's geometry as well as their production processes. I want to thank Oshaug Metall as well as GKN Aerospace for facilitating the fantastic opportunity of scanning the blade with a state-of-the-art optical metrology system.

I would also like to thank Professor Sverre Steen for help with explaining propeller theory, and useful hints on where to start on that particular topic.

Finally, I would like to thank my family for their support during my studies.

Trondheim, June 21, 2018

Njål Haagensli Munthe-Kaas

Table of Contents

Summary i

Sammendrag i

Preface ii

Table of Contents v

List of Tables vii

List of Figures x

1 Introduction 1

1.1 Background 1
1.2 Objectives 2
1.2.1 Reverse engineering 2
1.2.2 In-line quality control 2
1.3 Scope 3
1.4 Report structure 3

2 Propeller theory 5

2.1 Propeller classification 6
2.1.1 Fixed pitch propellers 6
2.1.2 Controllable pitch propellers 7
2.2 Propeller geometry 8
2.2.1 Propeller coordinate system 8
2.2.2 Propeller diameter 9
2.2.3 Pitch 12
2.2.4 Chord length 13
2.2.5 Skew and rake 13
2.2.6 Foil geometry 15

2.3 Propeller design basis 16
2.4 Propeller manufacturing 17
2.5 Standard on tolerances and measurements 18

3 Optical metrology systems for reverse engineering and in-line quality control 19
3.1 Reverse engineering 19
3.1.1 Geometry data acquisition 19
3.1.2 Digital reconstruction based on geometry data 21
3.2 Optical in-line quality control 21
3.3 3D scanners 22
3.4 Point clouds and data formats 24
3.5 Robotic scanning algorithms 24
3.6 Automatic alignment of point clouds 25

4 Methods 29
4.1 The propeller blade 29
4.2 Setup of robot cell with 3D scanner 30
4.2.1 Camera calibration 31
4.3 Automated scanning algorithm 32
4.3.1 Scanning of part table 32
4.3.2 Estimating the blade center axis 32
4.3.3 Generate first scanning poses based on initial guess of geometry . 33
4.3.4 Grid based search algorithm 33
4.3.5 Edge detection and classification 34
4.3.6 Termination of algorithm 36
4.4 Defining coordinate system 38
4.5 Extracting radial sections from the propeller point cloud 40
4.6 Opening out of cylinder sections 42
4.7 Calculating design parameters for each section 44

5 Results 49
5.1 Estimated design parameters with Zivid system 51
5.1.1 Pitch 51
5.1.2 Chord length 52
5.1.3 Skew and rake 53
5.1.4 Edge locations 54
5.2 Estimated design parameters from ATOS ScanBox 55
5.2.1 Pitch 55
5.2.2 Chord length 56
5.2.3 Skew and rake 57
5.2.4 Edge locations 58
5.3 Comparison 59
5.4 Sources of error 61

6 Conclusion 63
6.1 Discussion . . . 63
6.2 Recommendations for future work . . . 64
6.3 Final conclusion . . . 65

Bibliography 67

Appendix 71
A.1 Propeller Drawing . . . 71
A.2 Article accepted for SYROCO 2018 conference . . . 73
A.3 Code . . . 74

List of Tables

3.1 Comparison of different 3D scanners on the market 23

5.1 Estimated design parameters from scan with Zivid 51
5.2 Pitch error from Zivid scan 51
5.3 Chord length error from Zivid scan 52
5.4 Skew and rake error from Zivid scan 53
5.5 Edge location error from Zivid scan 54
5.6 Estimated design parameters from scan with ATOS ScanBox 55
5.7 Pitch error from ATOS scan 55
5.8 Chord length error from ATOS scan 57
5.9 Skew and rake error from ATOS scan 57
5.10 Edge location error from ATOS scan 58
5.11 Mean error and standard deviation of Zivid and ATOS scan 60

List of Figures

2.1 Propeller drawing with terminology 5
2.2 Controllable pitch vs. fixed pitch propeller 7
2.3 Propeller coordinate system 9
2.4 Propeller diameter 10
2.5 Cylinder section of a propeller blade 10
2.6 ‘Opening out’ of cylinder sections 11
2.7 Pitch of an opened out blade section 12
2.8 Skew angles on a highly skewed propeller blade 14
2.9 Skew and rake definition of a section 15
2.10 Foil geometry of a section 16

3.1 Data acquisition methods 20
3.2 A typical process of turning point clouds into a solid CAD model. 21
3.3 Structured light 3D scanner 23
3.4 Transformation matrices from 3D scanner to the robot’s base frame 26

4.1 Setup of robot with 3D scanner and digital twin 30
4.2 Estimation of sphere contain the propeller reference line 33
4.3 Normal parameterization of a line 35
4.4 Blade edge detection using Canny edge detector and Hough transform . . 35
4.5 Point cloud of the propeller blade 37
4.6 Angle of guide pin relative to the y-axis 38
4.7 Radial sections in point cloud 41
4.8 Segmented and opened out sections from point cloud 42
4.9 Opened out section at 0.7R 44
4.10 Finding TE and LE with the centroid 45
4.11 Section with added edges and chord line 45
4.12 Expanded profile from propeller drawing 46

5.1 Blade scanned with ATOS ScanBox 49
5.2 Comparison of scan quality at 0.4R 50

5.3 0.95R section from Zivid scan 52
5.4 Trailing edge detected at 0.7R and 0.8R 56
5.5 Comparison of design pitch and measured pitch 59
5.6 Comparison of design chord lengths and measured chord lengths 59

6.1 How the thickness of a section can be measured 63

Chapter 1

Introduction

1.1 Background

Measurement of the dimensions of ship propellers is a time-consuming and skill-demanding task. Compared with currently used measurement technologies, using a 3D scanner for optical measurements or metrology of propellers could prove to be a faster, simpler and fully automatic method. If implemented successfully, this would be of great benefit for the propeller manufacturing industry.

There has been little previous research on the topic of using 3D scanners for reverse engineering of propellers. Yeo and Choong [39] presents a method of characterizing the geometric properties of a marine propeller by utilizing a 3D scanner. This characterization methodology became a starting point for developing the reverse engineering system presented in this thesis.

Another article presenting reverse engineering techniques for propellers is [26]. Both of these articles provide useful inputs in regards to their methods of approaching the reverse engineering process for a propeller. However, they are both based on the reverse engineering of a fixed pitch propeller. To the author's knowledge, no scientific literature previously existed to describe the reverse engineering of a controllable pitch propeller. This is a more challenging task for reasons that will be outlined later in this report.

In general, reverse engineering of propeller blades is difficult compared to other objects. Due to the complex surface geometry, it is usually not feasible to do inline measuring of the propeller geometry during manufacturing. A potential 3D scanning process is further complicated by the highly reflective bronze surface of the propeller blades.

During design of propellers, some common design parameters are utilized to describe the geometry of the specific propeller. Trying to estimate these design parameters seems like the most useful way to approach the challenging reverse engineering problem for propellers.

1.2 Objectives

There are two different problems that could potentially be solved by the 3D scanning of a propeller blade.

1.2.1 Reverse engineering

One of them is reverse engineering of a propeller that has been in use, and where the data sheet or a CAD model for the propeller blade is not available. For example, for a ship propeller with 4 blades, if one of the blades has been damaged, one might want to scan one of the other blades to digitize the geometry data for the blade. This would allow the creation of a casting mould to manufacture a new blade.

1.2.2 In-line quality control

The other problem we want to look at the possibilities of solving in this project, is optical measurement or quality control of a propeller blade during the manufacturing process. In this scenario, the data sheet containing the geometry of the propeller blade is available, and the blade could be scanned with the intent of comparing the physical geometry with the design data.

Currently, when propeller blades have been cast, they are measured with a coordinate measuring machine (CMM) and then the excess material is manually grinded off. An automatic optical metrology system like the one presented in this project, could potentially be used in-line during the manufacturing of propeller blades. This sort of system could be placed in a robot cell with a grinding robot. Grinding paths for the robot could be calculated based on the results from the optical measurements. This has the potential to completely automate the process of grinding cast propeller blades down to the correct geometry. This is perhaps the most promising application of the method presented later in this thesis, in terms of reducing production costs for propellers.

The specific aims of this project is to

- Program a robot manipulator with a mounted 3D camera so that one can automatically take the required 3D images to scan a propeller blade. Generate a digital 3D model from the aligned scans.
- Present a solution for estimating the design parameters of the propeller based on the 3D model.
- Test the system in experiments. Evaluate the results.

This is to be done with the two scenarios - reverse engineering and in-line quality control - in mind. A good solution would be a solution that could be used to solve both problems.

1.3 Scope

In this thesis we will first and foremost focus on the process of calculating the design parameters from a point cloud of a propeller blade. The program that takes the point cloud of the propeller as input and estimates the design parameters of the propeller was written in MATLAB.

Originally, synthetic data from a CAD model of a propeller was produced in the form of a point cloud. Then, the procedure of estimating the design parameters of this point cloud was developed. An opportunity to test the procedure on a physical propeller was presented. A scanning algorithm was developed to produce the point cloud needed to test the system.

The steps that were made to acquire the aligned point cloud will also be described. An article describing the algorithm for automated scanning of the propeller blade was approved as part of this work, and can be seen in Appendix A.2.

The goal of the project is to develop and present a method of parameter-based reverse engineering of a single, controllable pitch propeller blade. The system could further be developed to work on all propellers, but it would need further testing and evaluation. The focus of the project was to develop a method to automatically estimate the parameters defining the section chords of a propeller blade. Thickness measurements of the sections are also a useful and interesting application of the 3D scanning of the blade, but this particular problem will not be the focus of this project.

1.4 Report structure

The report is organized as follows.

- In Chapter 2, the different design parameters that are used for the geometric design of ship propeller blades are described, as well as some general propeller terminology and classification.
- In Chapter 3, existing solutions for reverse engineering and optical metrology of propeller blades are presented.
- In Chapter 4, we outline the methodology used to produce the results.
- In Chapter 5, the results of the experimental process described in Chapter 4 are presented. The results are compared with theoretical data, as well as measurements from a state-of-the-art 3D metrology system. Sources of errors for the results are outlined.
- Chapter 6 contains a discussion about the project results, as well as a conclusion of the report. Recommendations for future work is presented, with suggestions on how the system can be improved.

Chapter 2

Propeller theory

Even though ship propellers all have the same purpose, to provide propulsion for a ship, there are thousands upon thousands of different types and designs of these propulsion systems. In this chapter, different types of propellers will be outlined, as well as the most important considerations for the design of the propeller and its blades.

Even though propellers can look very different depending on size, shape and function, they do share some common characteristics.

Figure 2.1 Drawing of a right-handed screw propeller with three blades [25].

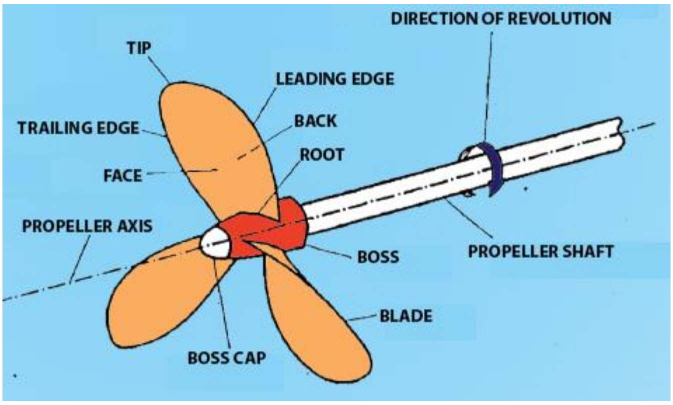


Figure 2.1 shows a diagram of a typical propeller with some common terminology for different parts of the propeller. Propellers usually consist of 3-6 blades that are mounted onto the boss or hub, with its center axis coinciding with the shaft axis. The propeller is called a right-handed propeller if it rotates clockwise seen from the aft when the ship is headed forward, and a left-handed propeller if it rotates counter-clockwise. The face and back sides of the blades will in this thesis be called the pressure and suction sides. The pressure side is the side facing the aft and leading into the water during rotation.

2.1 Propeller classification

There are many different types of propellers and ways of categorizing them. The type, size and geometric shape of a propeller is chosen based on a wide variety of different factors, such as the the type of ship it is designed for and the operating conditions the ship will be exposed to.

Ducted propellers, with the propeller placed inside a ring duct with a wing-like cross section, are commonly used for vessels with small draft, low speed and high thrust requirements, such as towing and trawling vessels [33]. This type of design offers protection to the propeller blades and provides additional thrust by controlling the water flow around the foil geometry of the duct.

Azimuthing thrusters are in frequent use for vessels with high positional maneuverability requirements, for example for floating production vessels in the offshore petroleum industry. The azimuth thruster arrangement provides an extra degree of freedom to the propeller, by allowing the propeller to rotate 360°around the vertical axis. An azimuth thruster can either be placed in a wing duct or have a non-ducted propeller arrangement [12].

Propeller blades can also be classified according to their geometric shape, such as the skew of the propeller. According to [5], a propeller is classified as a highly skewed propeller if it has a skew angle of more than 25°. How the skew of a propeller is defined, and what kind of impact this design consideration has on a propeller's operation, will be outlined later in this chapter.

The material of the propeller can also vary depending on a number of factors, such as for fresh water applications or whether the ship is built for operation in ice conditions. Most propellers designed for ice free salt water operation are cast in a type of copper alloy. The chemical composition of the standard copper alloys allowed for propeller blades and hubs are described in [19]. For large propellers, a type of nickel-aluminium bronze is commonly used.

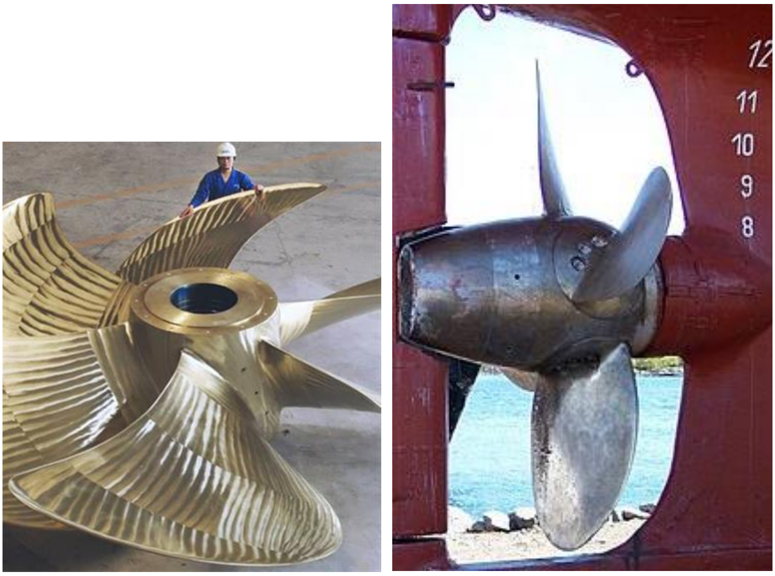
In this project, we are mainly concerned with how the propeller blades themselves are mounted to the propeller hub. There are two common ways this is done, making up another two major categories of propellers; Fixed pitch propellers and Controllable pitch propellers. These two types of propellers present different challenges in regards to reverse engineering or optical measurements of the propeller's geometry. In this subchapter, the characteristics of these two types of propellers will be outlined.

2.1.1 Fixed pitch propellers

A fixed pitch propeller is a propeller with its blades rigidly attached to the hub. The propeller can either be cast in one solid piece, or the blades can be cast separately and fixed with bolts to the hub. For fixed pitch propellers, the only operational variable is rotational speed. In Figure 2.2, an image of a fixed pitch propeller can be seen to the left.

Fixed pitch propellers are relatively cheap in production and maintenance, and is a popular choice of propeller for large ships that don't require precise positioning of the vessel. This includes tankers and container ships to a large extent. A drawback for fixed pitch propellers is that different types of stress on the blades can occur at different rotational speeds, and there is no way to adjust for this during operation. Thus, fixed pitch propellers are fitted on ships that will operate in a relatively constant RPM range. For

Figure 2.2 Two major types of propellers. Left: Fixed pitch propeller cast in a single piece. Image credit: [37]. Right: Controllable pitch propeller on a Hurtigruten ship. Image credit: [18].



highly skewed propellers of this type, applying full power astern can lead to noise and vibration potentially causing cavitation on the blades [5].

2.1.2 Controllable pitch propellers

A controllable pitch propeller is a propeller with a mechanism inside the hub which allows the blades to rotate around its own axes. This allows for remote operation to change the pitch of the propeller, thus providing an additional operational variable compared to a fixed pitch propeller. The hub diameter is usually larger for controllable pitch propellers compared to fixed pitch propellers because of this mechanism. An image of a controllable pitch propeller can be seen in Figure 2.2, to the right.

According to [12], some of the advantages of controllable pitch propellers include a higher degree of maneuverability, and fine control of the thrust. This is particularly useful in situations where dynamic positioning of the vessel is important, especially if the propeller is mounted on an azimuth thruster. The major drawback of this type of propeller is the complexity of the design, as well as the larger hub diameter meaning the total blade area of the propeller in many cases is reduced.

2.2 Propeller geometry

In this subchapter, the most important design parameters for ship propellers are outlined. In addition, their respective importance for the propeller's performance are described. The major geometric design parameters of propellers are, listed in approximate order of importance (from Sverre Steen, professor at Department of Marine Technology, NTNU):

1. Propeller diameter
2. Pitch and pitch distribution
3. Chord length distribution
4. Number of blades
5. Camber distribution
6. Thickness distribution

In addition,

- Skew
- Rake
- Hub diameter

could be added as important parameters, and ones we would like to account for in the reverse engineering process of this project. The book 'Marine Propellers and Propulsion' by John Carlton [12] has been used extensively for information on how the geometric parameters are defined.

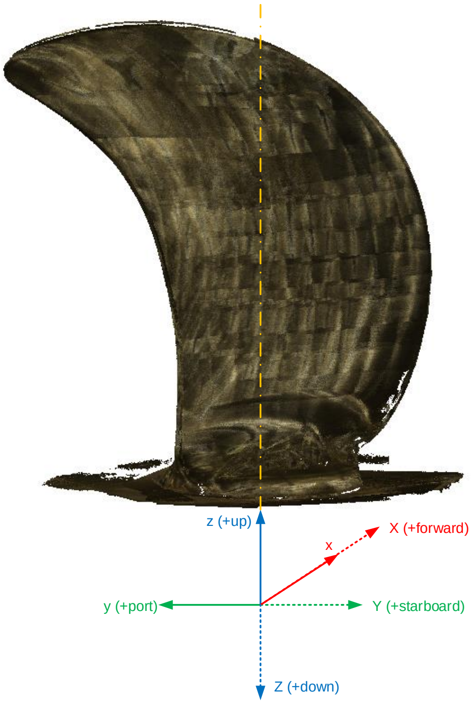
2.2.1 Propeller coordinate system

Both in the literature and in the industry, a wide variety of different definitions of the propeller reference frame are used. The lack of an internationally defined standard for the global reference frame of propellers is apparent with the industrial partner for this project, Oshaug Metall. The company receives propeller design data from customers in all types of different coordinate systems. They then convert the data to the coordinate system chosen as standard for the company.

Carlton uses a right-handed Cartesian coordinate system, as proposed by [34]. This global reference frame is defined so that the X-axis is positive along the propeller shaft axis in the forward direction of the ship. The Z-axis is positive vertically downwards, and the Y-axis is positive towards starboard of the ship.

In this project, it was useful to define a local coordinate system for the scanning of a controllable pitch propeller blade, where the z-axis is positive vertically upwards and the y-axis is positive towards port of the ship. The x-axis is still coincident with the shaft axis in the forward direction. The origin of the coordinate system is located in the center of the propeller hub, where the shaft axis intersects a line that runs through the center-point of

Figure 2.3 The global reference frame (XYZ) for the propeller geometry is shown as dashed lines. The local coordinate system (xyz) used for the propeller blade geometry in this project is shown as solid lines. The propeller reference line running through the center of the blade is shown as a dashed orange line.



the circular blade foot. This line is coincident with the z-axis, and is called the 'propeller reference line'. In this report the term 'blade center axis' is also used to describe this line. The global reference frame as well as the local coordinate system used in this project can be seen in Figure 2.3. In the figure the propeller reference line, about which the propeller blade is defined, can also be seen.

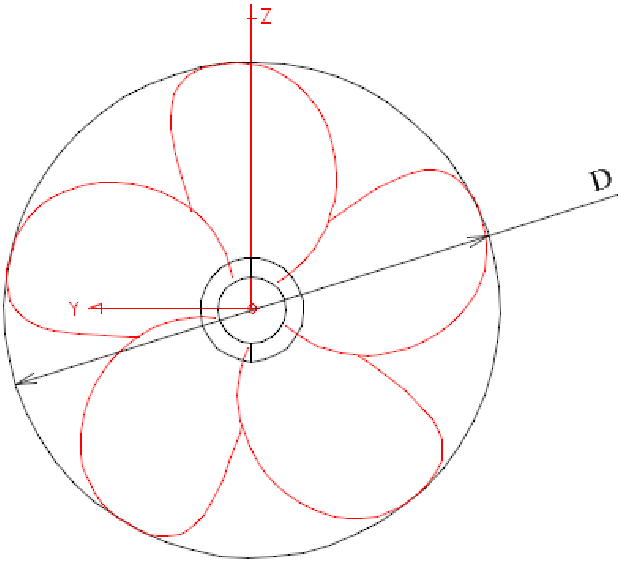
The coordinate system used by Oshaug Metall is again different from the system used in this project, with the z-axis positive in the forward direction, the y-axis vertically upwards, and the x-axis positive towards port.

2.2.2 Propeller diameter

The diameter of a propeller is defined as the diameter of the circle spanned by the propeller when it is rotating, seen from directly behind the ship. For a fixed pitch propeller, this is a relatively easy concept to understand. A drawing of the propeller diameter is shown in Figure 2.4.

For controllable pitch propellers the exact diameter is harder to determine, because the

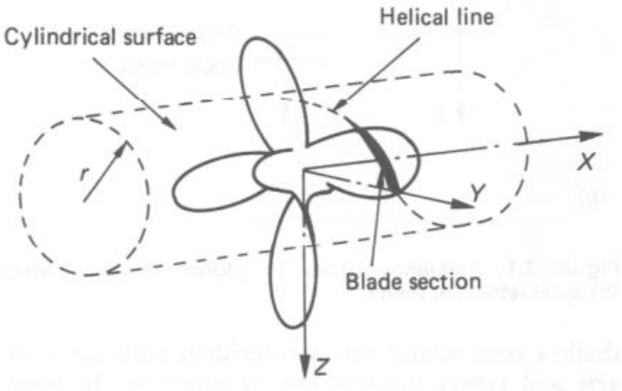
Figure 2.4 The propeller diameter (D) is defined as the largest span of the propeller in the YZ-plane [23].



circle spanned by the propeller varies depending on the pitch of the blades.

This is where the design parameters of the propeller comes in. The design parameters are given for radial sections of the blade at certain radii along the propeller reference line in the yz-plane. These radial sections can be understood as parts of the surfaces of cylinders with the specified radius and a cylinder axis coinciding with the x-axis. Such a cylinder section can be seen in Figure 2.5.

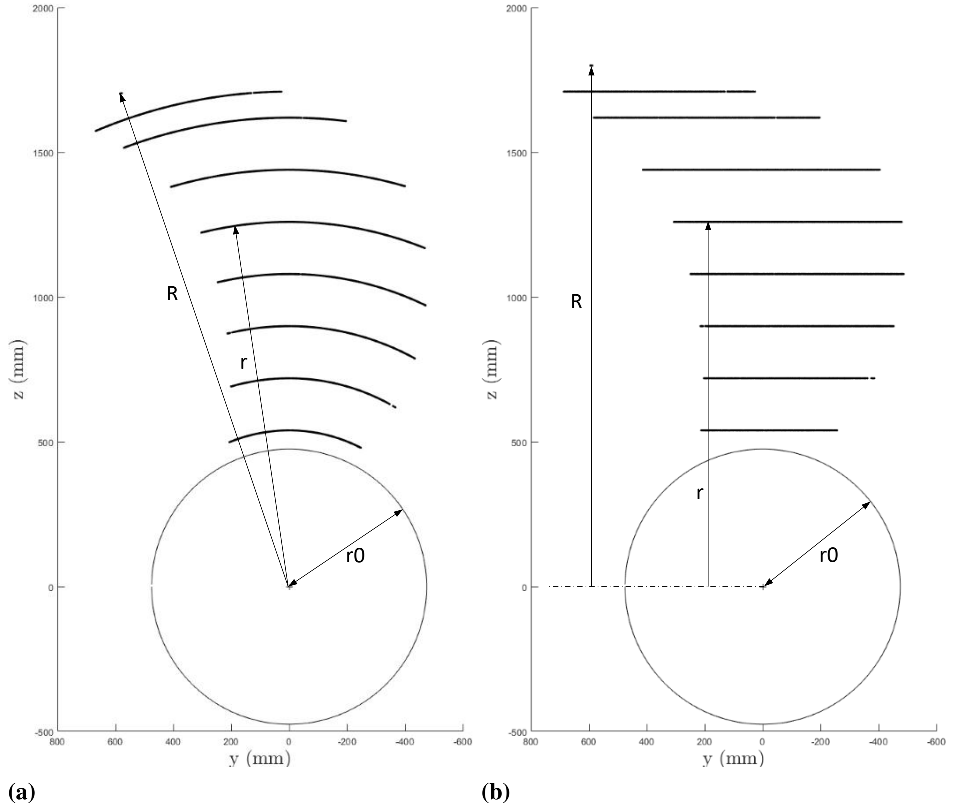
Figure 2.5 The radial blade section is a part of the surface of a cylinder cutting through the propeller blade [12].



The radii of the sections are given as fractions of the total propeller radius, $R = D/2$. For example, the section radii, r, can be given for $r/R = [0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95]$ of the total propeller radius.

The design parameters are defined for the blade sections when the cylinders have been 'opened out', as described by [12]. The surface of the cylinders are unwrapped, so that the cylinders are now rectangles along a flat plane. This process is shown in Figure 2.6.

Figure 2.6 'Opening out' of radial sections. The original cylinder sections of a propeller blade at different radii are shown in (a). In (b), the cylinders have been opened out so that the blade sections are defined along a flat plane. R is total propeller radius, $r = 0.7R$, and r_0 is the hub radius of the propeller.



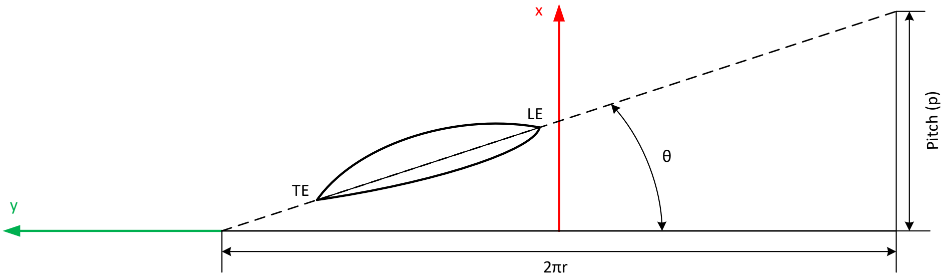
The blade sections are then laying along a plane parallel to the xy-plane, and with a distance from the xy-plane equal to the radial distance of the cylindrical sections from the origin, r. The opened out blade sections are also called the expanded blade sections.

For controllable pitch propellers, the concept of a 'design pitch' is used. The design pitch is defined for one of the expanded blade sections, usually at $r/R = 0.7$. The diameter of a controllable pitch propeller is found when the propeller blade is oriented so that the pitch at this specified blade section is equal to the design pitch.

2.2.3 Pitch

For each cylinder blade section, a helix can be developed through the leading and trailing edges of the section and around the cylinder it is a part of. This type of helical line can be seen in Figure 2.5. Carlton lists a number of different ways a helix can be defined, but in this project we will focus on the previously described definition, which is most commonly used in the propeller industry. The pitch can then be imagined as the distance the section travels in the x direction during one revolution around the cylinder along its helix.

Figure 2.7 An opened out blade section with a helix through the leading edge and trailing edge. The line segment between the edges is called the chord line of the section. The movement in direction x when the section travels along the helix defines the pitch of the section.



When the cylinder surface is opened out and projected on a flat plane parallel with the xy-plane, it might be easier to visualize this pitch distance. This process is seen in Figure 2.7. Here, the circumference of the cylinder $(2\pi r)$, is laid out flat along the y-axis, and the pitch distance p is seen as the movement of the blade section in the x direction when it travels along its helix (dashed line) for one full cylinder circumference.

The pitch for a section is also commonly given as an angle. The relationship between this angle (θ) and the pitch distance p is given by

$$\theta = \tan^{-1}\left(\frac{p}{2\pi r}\right) \tag{2.1}$$

This is a simple geometric relationship which has been used in this project to calculate the pitch of sections of a 3D scanned propeller.

Pitch is an absolutely crucial design parameter for a propeller, and perhaps the most important parameter after the propeller's diameter. The pitch angle determines the angle with which the blades attack the incoming water flow. This angle determines the pressure exerted by and on the blade, and so has a great impact on the efficiency of the propeller.

Each cylinder section of the blade has its own pitch, and a table describing the pitch for each blade section is what is called a pitch distribution. When designing propellers, it is desirable to balance the pitch distribution somewhat. Since the circumference of the cylinder surface cutting through the blade will increase as one moves further out on the blade, the pitch distance would increase if the pitch angle remains constant. The pitch distance is in fact the variable that is usually desired to stay relatively constant, and so the pitch angle is set as smaller further out on the blade than closer to the hub. This design consideration is what gives propeller blades their characteristic ‘twist’ shape.

2.2.4 Chord length

As seen in Figure 2.7, there is a line segment between the trailing edge and the leading edge of the propeller, which is aligned with the helix of the segment. This line segment is called the chord line of the propeller section. The length of this line, the chord length, is another important design parameter which is used by designers to describe the propeller's geometry.

The midpoint of this line is called the mid-chord. This point is commonly used to further define the rake and skew of the section. This will be further explained later in this chapter. In Figure 2.8, the developed mid-chord locus for the cylinder sections of a propeller blade is shown.

A table with the chord lengths at predefined radii of the blade is called a chord length distribution. From the chord lengths $c(r)$ at radii of a propeller blade, the total expanded area of the propeller can be expressed as

$$A_E = z \int_{r_h}^{R} c(r)dr, \tag{2.2}$$

where z is the number of blades, r_h is the radius of the root section, and R is the propeller radius. The blade area is a critical parameter, as it affects the propeller's efficiency and thrust to a large extent.

In the propeller drawings, the blade area is commonly given as a parameter called EAR, expanded area ratio. The expanded area ratio is a relationship between the area and diameter of a propeller, and is expressed as

$$EAR = \frac{A}{\pi R^2} \tag{2.3}$$

Increasing the area ratio of a propeller can reduce the risk of cavitation [33].

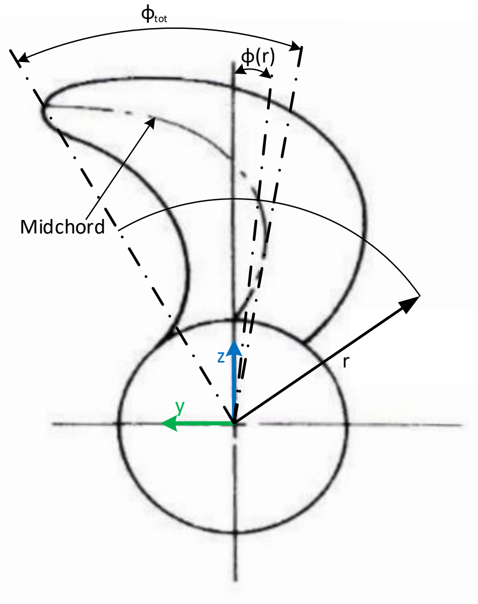
2.2.5 Skew and rake

We will use the definitions of skew and rake proposed by [12].

The skew angle for each cylinder section is the angle between the propeller reference line (z-axis) and a line passing from the origin of the coordinate system through the mid-chord of the section. In Figure 2.8, this angle is called $\phi(r)$.

The figure shows a typical highly skewed propeller blade. The purpose of this scimitar-like design is mainly to avoid cavitation, while not reducing the propeller efficiency in a major way [4]. Cavitation is caused by the pressure on the suction side of the blade becoming so low that the water starts boiling even at low temperatures. These vapour bubbles can lead to erosion of the blades, as well as noise and vibration reducing the efficiency of the propeller.

Figure 2.8 A typical highly skewed propeller blade. The skew angles of the blade are shown. $\phi(r)$ is the skew angle for a cylinder section at radius r. ϕ_{tot} is the propeller skew angle, which is the largest spanning angle between two lines passing from the propeller shaft axis to the mid-chord of propeller sections.



Continuing with the opened out view of a section in the xy-plane as in Figure 2.7, we have the pitch, skew and rake of a section all combined in Figure 2.9. The mid-chord is displaced in the y direction by $r\phi$, with the angle in radians. In drawings from propeller designers, the skew of the sections are often given as the displacement of the mid-chord from the x-axis along the helix. This skew distance, or simply skew, is then found by

$$\text{Skew} = \frac{r\phi}{\cos(\theta)} \tag{2.4}$$

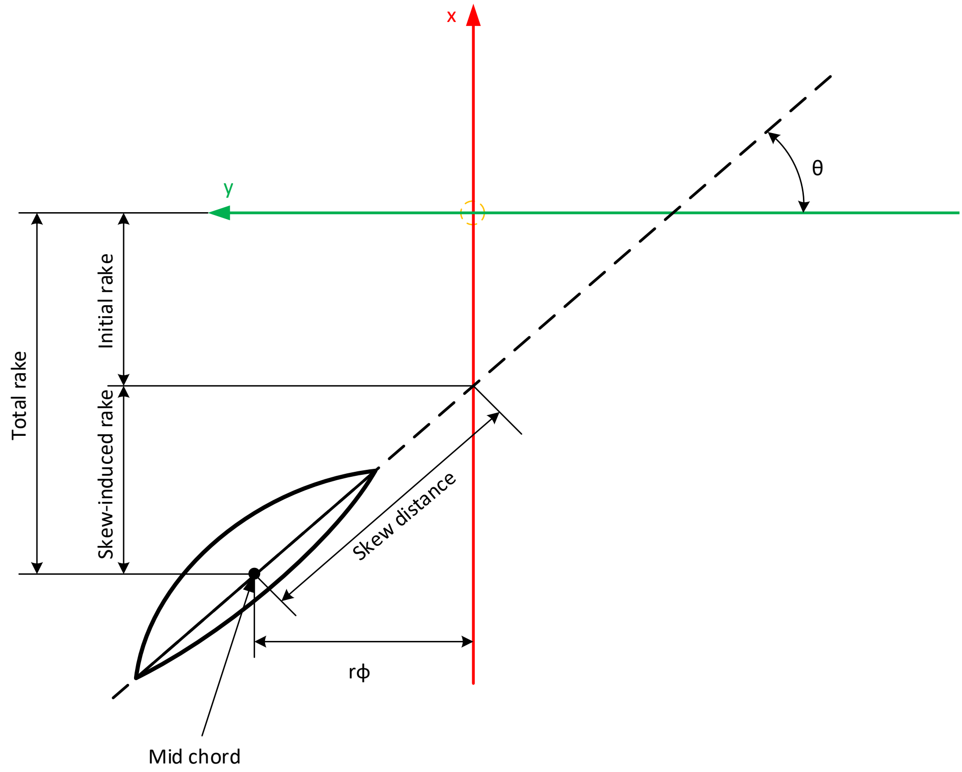
Also in Figure 2.9, we see the rake of the section. The rake is the displacement of the section along the shaft axis or x axis. It can be found as the distance from the propeller reference line to the mid chord in the x direction. If the mid-chord has a positive x value, the blade is angled forward. If it has a negative x value, it is angled backwards.

The rake can further be divided into two components: initial rake (i_{in}) and skew induced rake (i_s). The total rake (i_{tot}) of the radial section is then

$$i_{tot} = i_{in} + i_s \tag{2.5}$$

The initial rake is the distance from the propeller reference line to the point where the helix of the opened out cylinder section intersects the xz-plane. The skew induced rake is

Figure 2.9 Skew and rake of an opened out blade section relative to propeller reference line (aligned with z-axis, coming out of the plane).



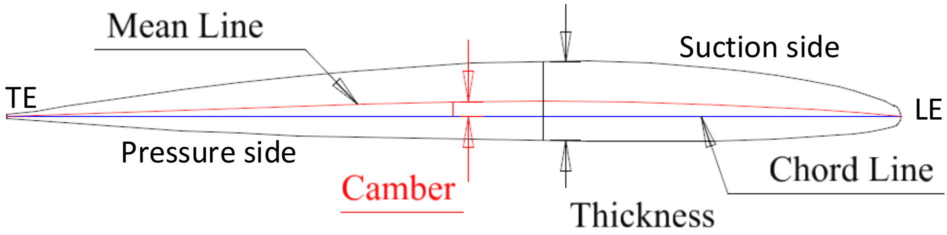
the additional rake that is caused by the skew of the section along the helix, and can be calculated by

$$i_s = r\phi \tan(\theta) \tag{2.6}$$

2.2.6 Foil geometry

The thickness of the blade sections is not equally distributed for each point across the chord line. The distance from the chord line to the suction side and pressure side, respectively, is also not necessarily equal at each point. This introduces a new concept regarding geometry of propeller blade sections, the camber. A so-called mean line can be drawn for the section, which is a locus of all the mid-points between the suction and pressure side of the blade. The mean line of a blade section can be seen in Figure 2.10. The distance from the chord line to the mean line is what is called the camber. The camber is measured on the mean line at points orthogonal to the chord line. If the camber is measured for several points across the chord length of the section, a camber distribution can be developed. This distribution

Figure 2.10 Camber and thickness measurements of an expanded blade section [23].



describes the wing-like shape of the blade section. Details of the most commonly used NACA series wing sections in propeller design are described in [2].

A thickness distribution is the measurement of the distance from the pressure side to the suction side at points of the chord line. This distance is also measured orthogonal to the chord. Many propeller designers operate only with a thickness distribution, as well as the distances from the chord line to each side of the blade, in their drawings. This is also what will be utilized in this project. But it is important to keep in mind the curved and unsymmetrical shape of the blade sections.

2.3 Propeller design basis

The term ‘propeller design basis’ is used when it comes to designing the propeller for a ship, and optimizing its geometry for that specific ship. There are three components to this term:

1. Required power
2. Rotational speed
3. Ship speed

These are the three most important operational factors that serve as the basis for the propeller design. These parameters need to be selected according to the mission profile of the vessel before any propeller geometry design can begin. In addition comes a wide variety of other factors, such as installation and running costs, maintenance requirements, legal requirements, as well as environmental factors like sea conditions and wind.

The geometric design parameters of the propeller can then be derived from standard series data and equations developed from years and years of research on the matter. Many computer programs exist that allow the user to input their operational factors and constraints, and generate the parameter based geometry of the best suited propeller, for example the free software OpenProp [14]. There also exists a wide variety of commercial software packages that can generate CAD models of propellers based on the input of the geometric design parameters.

2.4 Propeller manufacturing

In this subchapter, the current-day manufacturing process of controllable pitch propeller blades is presented. The manufacturing process includes several steps:

- Based on a CAD model of the propeller blade, a casting model is made.

  The industrial partner Oshaug Metall use automatically generated milling paths to create a wooden model based on the CAD model. The model is slightly oversized compared to the theoretical propeller design, to allow for shrinking during the contractions in the casting process, as well as a small margin of error.

- A casting mould in sand is made based on the wooden model.

  The wooden model is placed in a moulding box which is then packed with the sand to create the casting mould. 3D printing or automated milling of the sand casting moulds is another interesting research topic for the propeller manufacturing industry.

- The blade is cast in the mould.

  Casting of propeller blades is a very complicated metallurgical problem, especially when the goal is to cast the blade with as little excess material as possible. The varying thickness and shape of the blades means that air bubbles can occur, and when the cast cools down, the contraction of the metal can cause geometrical problems. In this particular project, we are not concerned with these casting challenges, and will assume a perfect cast of a blade.

- The blade is measured, and excess material is removed by manual grinding.

  Approximately 35-40 points on each side of the blade is measured with a coordinate measuring machine (CMM). The thickness of the blade at these points is then found. At this point, the thickness is hopefully equal to or greater than the nominal or theoretical thickness values at these points. Holes are then drilled at these points with the necessary depth to correct from the measured thickness down to the nominal thickness. After this, the blade is ready for grinding. An operator grinds the blade down to a state where the drilled holes at the measuring points are gone. In this process, one can only know for sure that the thickness at the measured points become correct. For every other point on the propeller surface, the exact thickness is unknown, and the quality control of the grinding process is based on the experience of the operator.

- The blade is polished and made ready for the customer.

2.5 Standard on tolerances and measurements

ISO484 is an ISO standard with the title Shipbuilding - Ship screw propellers - Manufacturing tolerances. The standard consists of two parts:

- Part 1: Propellers of diameter greater than 2.50 m (ISO 484-1:2015(E))
- Part 2: Propellers of diameter between 0.80 m and 2.50 m inclusive (ISO 484-2:2015(E))

The standard is used by propeller manufacturers to ensure that produced propellers comply with the manufacturing tolerances. It applies to both fixed pitch and controllable pitch propellers. The manufacturing accuracy is divided into classes S, I, II, III with S being the highest accuracy class and III being fairly wide tolerances. For a controllable pitch propeller in accuracy class I with a propeller diameter of 3600 mm, like the one scanned in this project, the mean pitch deviation allowed is $\pm 1\%$. The thickness measurements can deviate between $+2.5$ mm and -1.5 mm, or between $+2.5\%$ and -1.5%, whichever is greater. A problem with the standard is that these measurements only need to be done for the sections of the blade specified; Near the hub, 0.4 R, 0.5 R, 0.6 R, 0.7 R, 0.8 R, 0.9 R, 0.95 R. For all other points on the blade it will be unknown whether the blade has the correct geometry. The reason for this is that the current-day measuring systems can only measure one point at a time, so it would take a lot of time to measure for example 10000 points instead of 35 on each side of the blade.

The standard also describes the required accuracy of the measuring equipment: 'The maximum permissible inaccuracy of the measuring equipment shall not exceed half the tolerance on the dimension or quantity to be measured or, in the case of geometric measurements, 0.5 mm, whichever is greater.' The Zivid 3D camera used in this project has a specified depth resolution of 0.1 mm at a scanning distance of 0.6 mm, but it would need to be tested for accuracy to check whether it is accurate enough for this application.

Chapter 3

Optical metrology systems for reverse engineering and in-line quality control

3.1 Reverse engineering

Reverse engineering is a process of reproducing the geometry of an available physical object [36]. The general process of reverse engineering a part consists of two major steps:

1. Obtain as accurate as possible measurements of the geometry of the part.
2. Reconstructing the part as closely as possible to the geometry measured in step 1.

In this chapter we will look further into each of these two steps. We are particularly interested in two of the applications of reverse engineering that are described in [36]:

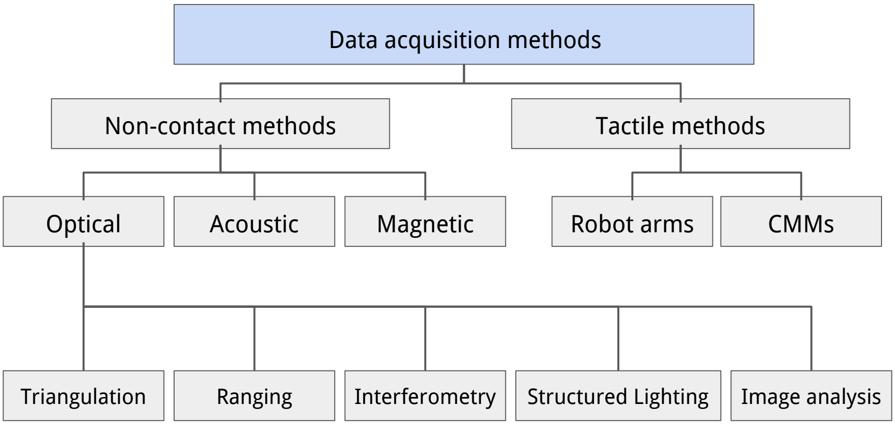
1. 'Old and overseas products which are not having prints and documentations can be produced.'
2. 'Automated inspection, the manufactured part is automatically inspected after a successful prototype is produced.'

3.1.1 Geometry data acquisition

Many different types of approaches and systems exist to acquire geometry data of a physical part or object. The most elementary approach is of course to measure the part manually, using calipers, line gauges, protractors or other equipment. For ship propellers, with their complex geometry and often substantial size, manual measurements becomes a challenging and time-consuming task. Thus, machine-based systems are more suitable for fast reverse engineering or quality control measurements of manufactured propeller blades.

Figure 3.1 shows a generic overview of systems that could be used to collect geometry data of a part for reverse engineering purposes. We see that measuring systems can be divided into two major categories; tactile methods, where a mechanical probe must touch the surface, and non-contact methods, where light, sound or magnetic fields are used to estimate the surface.

Figure 3.1 Methods and systems used for acquiring geometry data of a physical object, with a specific focus on optical measurement systems. Based on [36].



The system that is commonly used by propeller manufacturers today to measure the geometry of propeller blades, is coordinate measuring machines (CMMs). These machines have very high accuracy, but the measuring process is slow because they can only measure one discrete point on the surface at a time. Currently, CMMs are used for quality control of propellers during and after production. The propeller needs to be moved between workspaces and be clamped to the correct position for the CMM to be able to measure the correct points. Therefore these machines are not well suited for fast and automatic inline measurements during production.

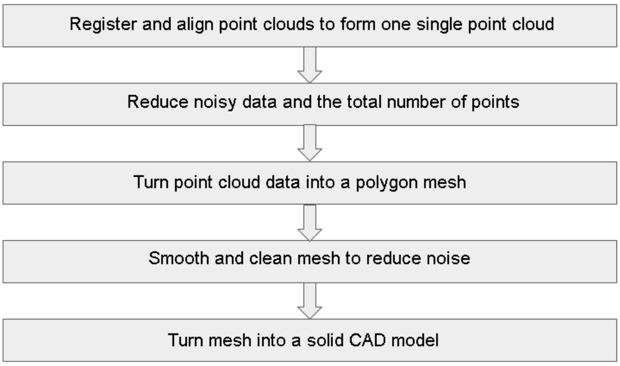
CMMs are also limited in their reverse engineering capabilities for propellers. It would take a tremendous amount of time and work for a CMM to measure as many points as would be needed to completely and accurately reverse engineer a propeller blade. Therefore, we are interested in looking at the possibilities of utilizing other systems for both the reverse engineering problem and potentially for inline quality control of manufactured propeller blades.

In this project, a structured light 3D scanner mounted on a robot manipulator will be used to investigate the opportunities of this type of system being used for both reverse engineering and quality control purposes of propeller blades. The scanner can capture millions of points on the surface of a propeller blade at a rate of 10 Hz. More on this type of scanner and the type of data it captures will be presented later in this chapter.

3.1.2 Digital reconstruction based on geometry data

Often, the goal of this part of the reverse engineering process is to create a CAD model of the object with the geometry data from part 1. The typical process steps of turning a cloud of measured points into a solid CAD model is shown in Figure 3.2.

Figure 3.2 A typical process of turning point clouds into a solid CAD model.


Register and align point clouds to form one single point cloud
Reduce noisy data and the total number of points
Turn point cloud data into a polygon mesh
Smooth and clean mesh to reduce noise
Turn mesh into a solid CAD model

In the case of a propeller blade specifically, one approach to the reverse engineering process could be to try to estimate the so-called design parameters of the propeller. Then, plenty of solutions already exist to create a CAD model of the propeller based on the input of these design parameters and further interpolation of discrete points on the surface. For the purpose of in-line quality control as well, there may be no need to go through the steps of turning the point data into a CAD model. Therefore, we are most interested in working with measurements in the point data itself in this project, but it is important to keep in mind that the goal of reverse engineering often is to obtain a CAD model. For the generation of automated robot grinding paths as well, there may be a need to reverse engineer a CAD model of the propeller blade.

3.2 Optical in-line quality control

The term in-line quality control refers to the measurement of a part or object during the manufacturing of said object. The goal is to quantitatively determine the physical geometry's deviation from the design geometry. We want to investigate the possibilities of using an optical metrology system, or 3D scanner, for in-line quality control of propeller blades.

One of the major advantages of using an optical metrology system for in-line quality control is these systems' measurement speed [31]. The current measurement technique for propeller blades utilizing CMMs is time-consuming and requires manual labour for clamping the propeller blade to the correct position and movement of the blade between production locations. If implemented correctly, a blade would not require careful positioning and clamping to acquire good measurement results from an optical metrology system.

Large ship propellers are manufactured in small series, and with the geometry of the propeller optimized for the ship it is designed for. A particular design might only be manufactured one time. Thus, another potential advantage of optical metrology systems for the propeller manufacturing industry is that these scanners assume no prior knowledge of the part geometry. Such a quality control system could potentially be completely automated. The results from the scan could be compared with the nominal design values, and the system could automatically notify a grinding robot of how much material needs to be grinded off at each point. Due to the wide variety of geometries produced by propeller manufacturers, the 3D scanner used for in-line quality control would need to be mounted on a robot manipulator, with the robot poses where scans are made generated automatically. This is to make the system adaptable and robust from the smallest to the largest propeller produced.

3.3 3D scanners

In this project we will focus on using a 3D scanner to generate a digital representation of a propeller blade. The terms 3D camera and 3D scanner will be used interchangeably in this report, as a 3D camera is simply an optical 3D scanner. 3D scanners fall into the non-contact category of measurement systems, with most of them operating with optical sensors, i.e. cameras.

3D scanners can generally be divided into two categories, outlined by [30]:

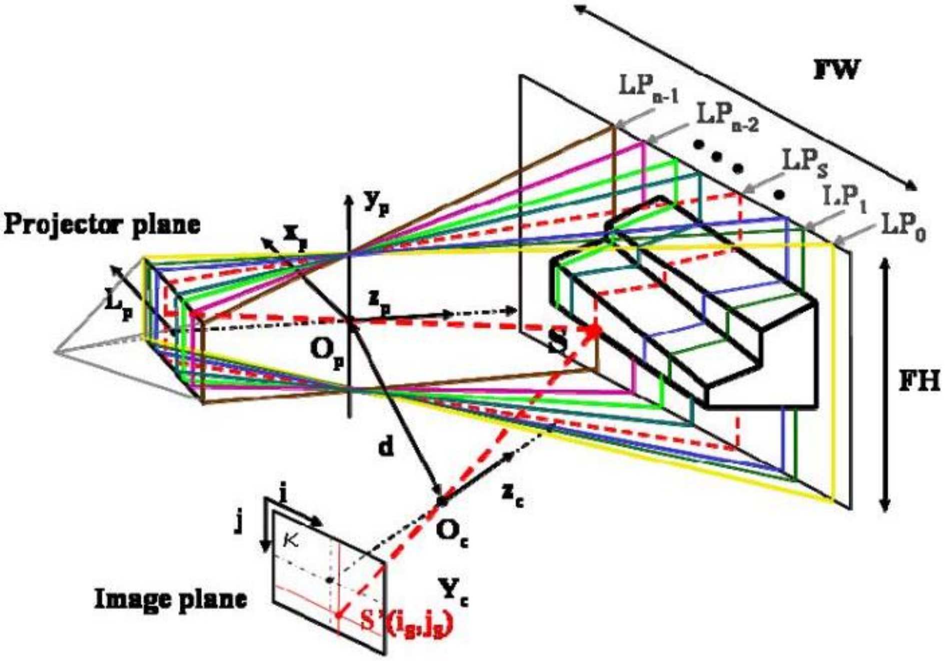
- **Active**: Scanners that operate by projecting electromagnetic energy onto an object and and recording the energy transmitted on the object.
- **Passive**: Scanners that acquire the electromagnetic energy that is reflected off an object.

The sensor used in this project, Zivid, is a type of 3D camera that falls into the active category of 3D scanners, specifically utilizing a structured light sensor. A structured light sensor uses a triangulation technique to capture point data. In Figure 3.3 we see an illustration of this triangulation technique.

A projector produces a pattern of light which is transmitted onto the object. An example point S shows how the light reflected off the surface of the object can be translated to the point S' in pixel coordinates in the image plane. The origin of the camera's coordinate system is in O_c. The X and Y coordinates of the point S are found by the (i, j) pixels in the image plane and the camera's intrinsic parameters. The depth data of the point, or the Z coordinate, is obtained by calculating the point where the line from S' to S intersects the plane LP_S.

Table 3.1 shows a comparison of different 3D scanners currently on the market. We see that there exists 3D scanning systems in many price ranges. Generally, the more expensive systems have a greater degree of accuracy. A 3D scanner like the Microsoft Kinect, with a depth accuracy of 10 mm will not be sufficiently accurate for reverse engineering and quality control applications on propellers. The Zivid 3D scanner used in this project has a listed accuracy two orders of magnitude more accurate than the Kinect.

Figure 3.3 Principle behind a structured light 3D scanner. From [30].



3D scanner	Approx. Price	Resolution	Operation range	Depth accuracy
MS Kinect [3]	$150	640x480 px	0.8 - 3.5 m	10 mm @ 2m
Sense [1]	€349	1920x1080 px	0.2 - 1.6 m	1 mm @ 0.5m
Zivid [43]	$10,000	2.3 Mpxs	0.6 - 1.1 m	0.1mm @ 0.6m
Creaform Handyscan 700 [13]	$50,000 -100,000	1.375 Mpxs	0.1 - 4 m	0.03 mm

Table 3.1: Comparison of different 3D scanners on the market

A complete optical 3D metrology system called the ATOS ScanBox [20] is made by German company GOM. The system looks to be the most complete and accurate optical 3D measurement system for industrial applications on the market today. It includes a structured light sensor mounted on a robot arm, with automatically generated scan poses. In many ways, the system is similar to the one developed in this project. One difference is that with this system, marking stickers need to be placed on the blade surface or its surroundings for alignment of scans. The ScanBox also includes comprehensive software that allows for parametric analysis of turbine or propeller blades, as well as the possibility to create CAD-models of the measured object. Test measurements with this system was made on the same propeller blade that was scanned in this project. Discussions of the results of these tests are found in Chapter 5.

3.4 Point clouds and data formats

The Zivid 3D scanner captures point data in the form of point clouds. A point cloud is a large set of points captured by a 3D scanner and defined by their position in a X, Y, Z coordinate system. The point cloud can also contain R, G, B and intensity data of the points if they are captured by a colour 3D camera. In some cases the surface normal vectors $n_i = (n_x, n_y, n_z)^T$ are also included in the point cloud data [16]. The point cloud usually represents the external surface of an object.

Many different file formats used to represent point clouds exist. Any format that can store 3D data either as points, vertices or polygons can be used for this purpose. The point cloud file formats that have been used in this project are as follows:

- Zivid Data files (.zvd): The format used to store point clouds captured by the Zivid camera.

- Polygon File Format (.ply): File format designed by Stanford graphics lab to store three dimensional data from 3D cameras.

The .zvd files are converted to .ply because this is an easy format to read in for example MATLAB. The location data in the .ply-file is called an unorganized point cloud, with the points as a 3-by-M matrix, where M is the number of points, and the columns represent the $[x, y, z]$ values for the points. This matrix is used to calculate the design parameters of the propeller blade later in this thesis.

3.5 Robotic scanning algorithms

As previously mentioned, we are interested in mounting the 3D scanner to a robot manipulator in this project, to be able to develop a method that could be used to automatically scan propellers with all kinds of sizes and geometries. Various research has previously been done regarding automatic 3D scanning of objects with robot manipulators.

An automated system for acquisition of complete 3D models is presented in [10]. The system, called RoboScan, utilizes a laser-based 3D scanner mounted on an industrial robot. A user must define the bounding volume of the scan object, and this is used to automatically generate the robot scan poses. Merging of the range maps caught with the scanner is done by exploiting the 3D scanner's positional data in the robot's workspace.

Larsson and Kjellander [24] focuses on path planning for automatic scanning of objects with curved surfaces. The equipment used in this paper includes a laser profile scanner, a robot manipulator and a turntable to provide an extra degree of freedom for the system. This system also relies on the user input of some parameters regarding the object's size and positioning. However, it is able to adapt to the object's shape and find the viewing directions for scanning which produces the most accurate results.

A solution to the problem of visual servoing for robot manipulators based on uncalibrated 3D images is presented in [9]. A 3D camera model is used to map positions in the robot workspace to a visual feature space, as well as computing a full-rank Image Jacobian Matrix. When it comes to the topic of robotic 3D scanning, such a system can help avoid collisions with the scan object, visual occlusion, as well as robot singularities, if it is combined with the part's geometry data and the kinematics of the robot.

Similarly to the system used in this report, [38] also presents a 3D metrology system using a structured light vision sensor mounted on a robot manipulator. To improve the accuracy of the structured light sensor a new approach to calibrating the sensor is proposed. This approach is based on determining the real projected centres of several fixed concentric circles manufactured in a calibration target. This is an interesting approach to keep in mind if the accuracy of the scanner calibration implemented in this project is unsatisfactory.

The scanning systems described in the articles earlier in this subchapter all require some knowledge or user input about the geometry of the measured object. Some assumptions about the scan object are also made in this project. The fact that it is a controllable pitch propeller blade with a propeller reference line vertically upwards through the centerpoint of a circular blade foot is one of these assumptions. Beside from that, no assumptions about the size and shape of the propeller blade is made, other than that the reach of the robot is large enough to acquire scans in all directions.

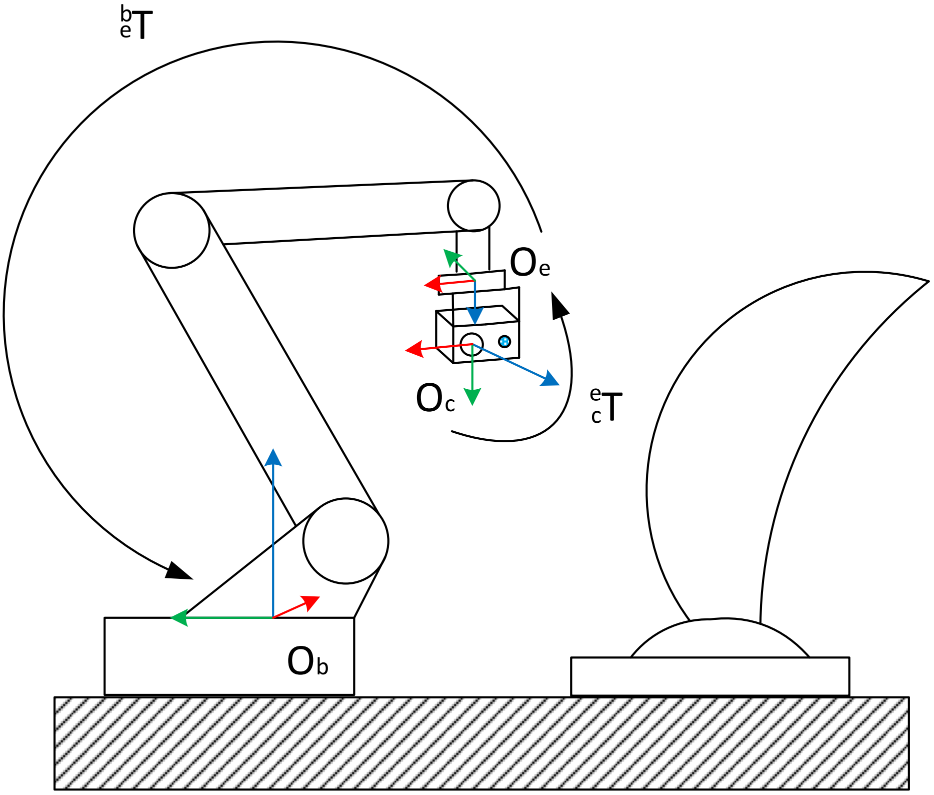
The approach for an automatic 3D scanning method of a propeller blade proposed in this project, is presented in chapter 4. Localization of the propeller blade in the robot base frame, a brute-force grid-based search algorithm for scanning, as well as automatic alignment of point clouds is included in the method.

3.6 Automatic alignment of point clouds

When acquiring multiple 3D scans of an object from different poses, the problem of aligning the point clouds arises. The points need to be accurately converted to the same coordinate frame to allow the creation of a complete 3D model of the object. There are several methods that can be used to achieve this. For reverse engineering purposes, when a 3D model of the object is not available prior, a commonly used method is called the Iterative Closest Point (ICP) method, first introduced in [6]. As the name suggests, this method is an iterative algorithm where the goal is to find the minimal translation and rotation between the points of two different point clouds. Some of the disadvantages of the ICP algorithm is that the algorithm needs an initial guess of how the point clouds align, and the process is relatively slow. We are therefore interested in solutions that can be completely automated more easily.

In this project, the process of aligning point clouds was automated by using the robot's base frame to establish a reference coordinate frame for all the scans. The problem of aligning the scans was overcome by using the positions of the robot end joint to calculate each scan pose in the reference coordinate frame. However, the 3D scanner has a displacement, and thus a different coordinate frame relative to the end frame of the robot. This is where calibration of the 3D camera comes in. The goal of the calibration is to calculate the extrinsic camera parameters, or the translation and rotation from the camera's coordinate frame to the end joint's coordinate frame. The translation and rotation can be defined in a single 4-by-4 matrix called the homogeneous transformation matrix.

Figure 3.4 Schematic drawing of the use of the transformation matrix between the camera frame and the end frame, and the transformation between the end joint and the robot's base frame to convert all scans to a single coordinate system.



The transformation matrix $^{e}_{c}T$ between the camera frame and the end frame can be written as, based on [32]:

$$^{e}_{c}T = \begin{bmatrix} R^e_c & t^e_c \\ 0^T & 1 \end{bmatrix} \tag{3.1}$$

where $t^e_c = [x, y, z]^T$ is the translation in x, y and z directions from the origin O_c of the camera frame to the origin O_e of the end joint. The 3-by-3 rotation matrix R^e_c is

$$R^e_c = R_z(\psi)R_y(\theta)R_x(\phi) \tag{3.2}$$

where

$$R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

is the rotation of an angle ψ about the z-axis,

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.4}$$

is the rotation of an angle θ about the y-axis and

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.5}$$

is the rotation of an angle ϕ about the x-axis.

Figure 3.4 shows a drawing of how transformation matrices are used to map positions in the camera frame to the robot base frame. The position of a point P_b in the robot base frame is calculated by

$$P_b \quad = \quad {}^b_eT \times^e_c T \times P_c \tag{3.6}$$

where P_c is the same point in the 3D camera's coordinate frame, and ${}^b_eT$ is the transformation matrix from the robot end joint to the robot base frame.

A similar method of mapping points from a sensor frame with different poses to a robot manipulator's base frame is presented in [40]. The measuring equipment used in this paper is a linear laser sensor and a precise linear rail mounted on the robot end link. However, the process of using the transformation matrices between the laser sensor and the robot base frame to align several point clouds of a car door panel, is very similar to the method used in this project. A hand-eye calibration method to determine the transformation between the sensor and the end joint is also presented in the paper.

Chapter 4

Methods

Referring to the objectives stated in the introduction, there are two goals that solutions should be presented for in this project. The first is to automatically generate a digital 3D model, in this case a point cloud, of a propeller blade. The second objective is to estimate the design parameters of the propeller blade based on this point cloud. The methods used to solve these objectives will be presented in this chapter.

4.1 The propeller blade

The propeller blade scanned in this project is designed for a controllable pitch propeller with a total propeller diameter of 3600 mm. The blade is right-handed and cast in Ni-Al-Bronze. It was manufactured by Oshaug Metall AS, where they use this blade as a master for calibration of their CMM. The blade geometry is therefore assumed to be fairly accurate compared to the nominal design geometry. It is manufactured in accordance with the tolerances in ISO 484-1:2015(E), accuracy Class 1. The blade has a skew angle of 29°, thus it is categorized as a highly skewed propeller. The blade is to be mounted on a propeller hub, along with three equal propeller blades. The propeller blade can be seen in Figure 4.1a, along with the robot and 3D scanner used for the automated scanning.

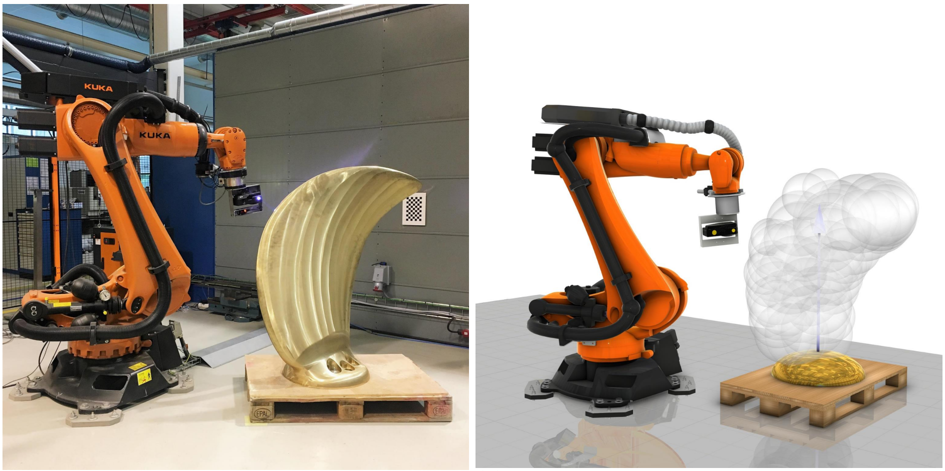
After the casting and grinding process the blade surface was polished. Reflections from these bronze propeller blades could be a major issue with regards to the accuracy of the optical measurements. However, the propeller blade has been in storage for some years, and due to oxidation it might not be as reflective as immediately after production. The reflectivity of the material is something to keep in mind when looking at the quality of the scans. The propeller drawing with all the design parameters of the propeller can be found in Appendix A.1.

4.2 Setup of robot cell with 3D scanner

A system for automatic scanning of the propeller blade was set up in a robot cell. The system contains the following components:

- Zivid 3D camera with High Dynamic Range (HDR) imaging capabilities:
  A structured light 2.3 Mpixel 3D RGBD camera, with a field of view of 425×267 mm and a depth resolution of 0.1 mm at an image distance of 0.6 m [42]. The optimal working distance of the camera is stated as between 0.6 and 1.1 m.
- Sick DT-35 Laser distance measurement sensor.
- KUKA KR 120 Industrial robot manipulator with its associated KRC4 robot controller.
- A central computer which processes the point clouds resulting from the 3D camera, generates the bounding spheres of point clouds, performs in-the-loop virtual simulation of the robot movements with reach analysis and collision detection, and controls the robot movements.

Figure 4.1 The physical scanning setup is mapped to a virtual simulation environment. The simulation is always one step ahead of the real world environment to detect and prevent collisions.



(a) Setup of robot with 3D scanner and propeller blade.

(b) Minimal bounding spheres generated from point clouds of scans.

The setup is shown in Figure 4.1a. For each point cloud the 3D scanner captures, a bounding sphere of the points in the scan is estimated. A bounding sphere is a sphere with the smallest diameter that can fit around all of the points in the point cloud. The bounding spheres are automatically added to the virtual simulation environment Visual Components 4.0 [27]. Figure 4.1b shows the simulation environment with the generated bounding spheres.

In this figure we also see the blue center axis of the blade that is automatically calculated. The method with which the center axis is estimated, and how the whole blade is automatically scanned from this, will be outlined in Chapter 4.3. How the point clouds are aligned to one global reference frame will also be outlined there.

4.2.1 Camera calibration

The calculation of the extrinsic camera parameters, or the transformation matrix ${}^{e}_{c}T$ between the camera frame and the end frame of the robot, was done using the RGB data from the 3D camera, and not the point coordinates. Images of several views of a chessboard pattern was taken with the scanner mounted on the robot, and the extrinsic parameters were calculated with functions from the OpenCV camera calibration module [8]. The intrinsic camera matrix of the Zivid 3D scanner was also used as part of the calibration, and it was assumed that the intrinsic factory calibration of the scanner is good and accurate. The camera calibration technique in the OpenCV library is based on [41] and [7].

4.3 Automated scanning algorithm

An overview of the steps included in the automated scanning algorithm is presented below:

1. The pallet which the propeller is bolted onto is scanned.
2. The propeller reference line is estimated using an algorithm called MSAC.
3. An initial guess for the propeller geometry is used to generate the first scanning pose.
4. Different scan poses of the propeller is found by using a grid based algorithm and data from the laser distance sensor.
5. Detection and classification of edges in the images are used to determine the direction of the step between each robot pose.
6. The algorithm is terminated when two edges with different angles are detected in one scan.

Each step of the scanning process is explained in detail below.

4.3.1 Scanning of part table

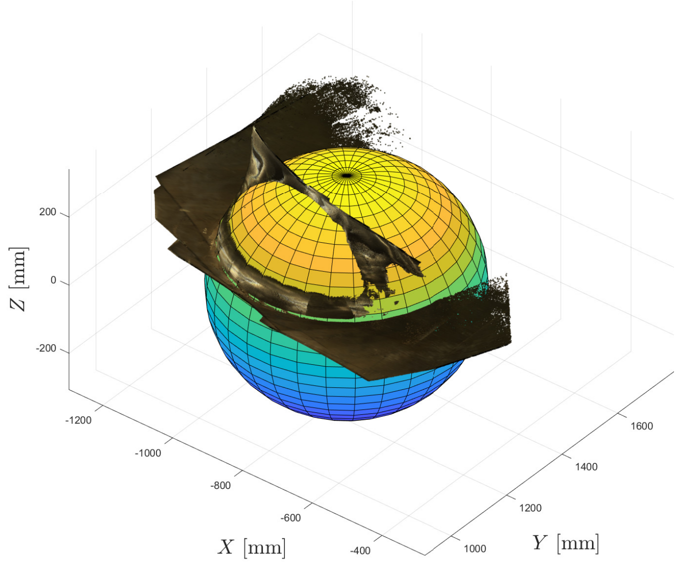
The position of the table on which the propeller is placed, as well as the scan poses necessary to scan this table is assumed to be known, if the method is implemented in an industrial setting. All kinds of propellers could then be placed on this table, and the initial estimation of the blade geometry is based on the blade foot which will appear in the scans of the table. The robot poses needed to scan the pallet the propeller is mounted on was hard-coded in this project.

4.3.2 Estimating the blade center axis

To generate further scan poses of the propeller, as well as avoiding collisions when scanning, it is necessary to estimate where the propeller is located in the robot's workspace. In this project, the approach chosen to do this included to first estimate the blade center axis, or propeller reference line, in the blade foot. From the aligned scans of the table the propeller is mounted on, an algorithm called M-estimator SAmple Consensus (MSAC) [35] is used to find the center of the blade foot. This algorithm is based on the RAndom SAmple Consensus (RANSAC) algorithm introduced in [17]. The MSAC algorithm looks for sphere-shapes in the point cloud of the pallet. Figure 4.2 shows a part of the blade foot estimated as a part of a sphere.

This sphere is then used to define the blade center axis, which runs vertically upwards through the blade foot. The center point of the sphere is assumed to be a point along this line. The blade center axis added to the simulation environment as a blue line is shown in Figure 4.1b.

Figure 4.2 The propeller blade center axis and radius of the blade foot is found by using MSAC to estimate a sphere from an initial safe-distance scan of the part positioning table. The sphere center point and radius are then used to set the starting pose for the propeller blade scanning.



4.3.3 Generate first scanning poses based on initial guess of geometry

From the blade center axis, an initial bounding box of the propeller is estimated as a cylinder along the axis with a radius equal to the radius of the sphere detected with MSAC. The first generated scanning pose is a point located at a good distance from the estimated blade center axis, at least $0.6 + r$ m from the line, with 0.6 being the lower bound of the optimal working distance of the camera and r being the radius of the sphere. The pose is calculated to be in a position where some of the edge of the pallet, as well as the blade foot, is visible in the field of view of the camera. The principal camera axis, or the z-axis, as well as the x-axis, is horizontal and parallel to the floor in the first scanning pose. The rest of the blade is then scanned from the bottom up, by moving from side to side in a grid.

4.3.4 Grid based search algorithm

A brute-force search algorithm is implemented to scan the rest of the blade. The algorithm generates a mapping grid around the cylinder representing the bounding box of the propeller, with a step length between the cells in the grid of around 15 cm. The algorithm starts with the initial scan pose found in the previous step, and moving the 3D scanner stepwise from right to left. For each scanning pose, information from the laser distance sensor is used to check the distance from the scanner to the blade surface. The scanner is then automatically moved along its z-axis so that the distance is now 0.6 m from the surface, which should be the distance which produces the most accurate scan results. Then, when the algorithm detects an edge in a scan, the scanner is moved up a cell in the grid, instead of continuing sideways. The following scan poses are then found moving the scanner in the opposite direction to the previous row, now left to right. How the algorithm detects and classifies edges is described Section 4.3.5.

As previously mentioned, for each scan a bounding sphere for the point cloud is calculated, and the estimation of the blade geometry and location is updated. The best guess of the geometry would be based on the bounding sphere with the smallest possible radius, but finding the minimal bounding sphere is computationally expensive. A faster solution is wanted with the speed of the scanning algorithm in mind. In this project, a near-optimal bounding sphere is found for each point cloud, utilizing the algorithm from [29]. The near-optimal bounding sphere is only about 5% larger than the minimal bounding sphere, and provides a good enough estimation of the propeller geometry to be able to perform collision detection and reach analysis during the scanning process.

With the bounding spheres added to the virtual simulation environment Visual Components, this program allows for automatic collision detection and reach analysis for the robot during the scanning. The simulation is also linked with the robot controller via an API, so the robot can be programmed and moved within this digital environment. The simulation of the robot's movement is one step ahead of the real world robot. The collision detection and reach analysis is performed before each scan, not allowing the robot to move before the simulation has verified that the robot can move unobstructed to the next pose. If the program detects that the robot can not reach a certain pose, the orientation of the scanner around its principal camera axis can be relaxed, so the scanner doesn't have to be aligned parallel to the floor.

4.3.5 Edge detection and classification

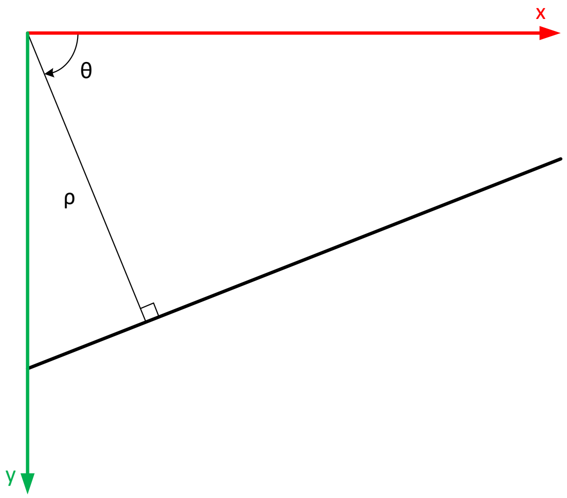
In addition to the depth and colour data associated with the point clouds, the Zivid 3D scanner also outputs a 2D image, also known as a pixel intensity image, for each scan. Using these images, the edges of the propeller blade can be found. The Zivid camera itself is pretty useful for segmenting out the background when scanning. Points that are far enough away from the camera are not included in the point clouds or intensity images (i.e. much of the background has depth values over a certain threshold value, and is removed). Thus, in the intensity images of the edges of the blade we get a nice contrast between the blade surface and the null values, showing up as black pixels. This ensures that in the images the edges are rather easy to detect using the correct method.

In this project, a Canny operator [11] was implemented for the edge detection in the intensity images. The edges in the images are almost always curved. We want to detect the edges to classify them as either side edges or top edges. The method of doing this involved using a Standard Hough Transform [15] to find straight lines in the curved edges. The Hough transform uses the so-called normal parameterization of lines to describe the geometry. This parameterization can be specified by

$$\rho = x \cos\theta + y \sin\theta \tag{4.1}$$

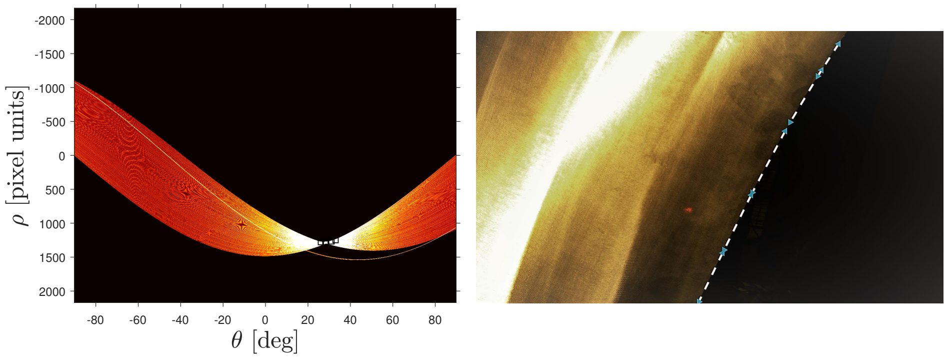
where ρ is the algebraic distance of the line from the origin in pixels, θ is the angle of the line normal, and x and y are the pixel coordinates. How these parameters are defined for a line is illustrated in figure 4.3.

Figure 4.3 The normal parameterization of a line used in the Standard Hough Transform.



The Hough matrix showing the relation between ρ and θ for each pixel along the detected edge is shown in Figure 4.4a. At the peaks in the matrix, where the parameters for many pixels along the edge are the same, the pixels are defined to lay on the same straight line. The peaks where this occurs in the matrix is shown as squares. The corresponding straight line segments in the intensity image is shown in Figure 4.4b.

Figure 4.4 Blade edge detection using Canny edge detector and Hough transform



(a) The Hough matrix for an image. Peaks marked as squares represents straight lines detected.

(b) The edge detected in the image, with the straight line segments marked on it.

The edges are classified as a side or top edge based on the angles of the straight lines found on the edge. If $|\theta| < 45°$, the line is more vertical than horizontal. If the lines detected on an edge is more vertical than horizontal in general, the edge is classified as a side edge, meaning that the scanning algorithm will continue by stepping up to the scanning pose or cell in the grid map above the current cell. If $|\theta| > 45°$, the edge is classified as a top edge.

4.3.6 Termination of algorithm

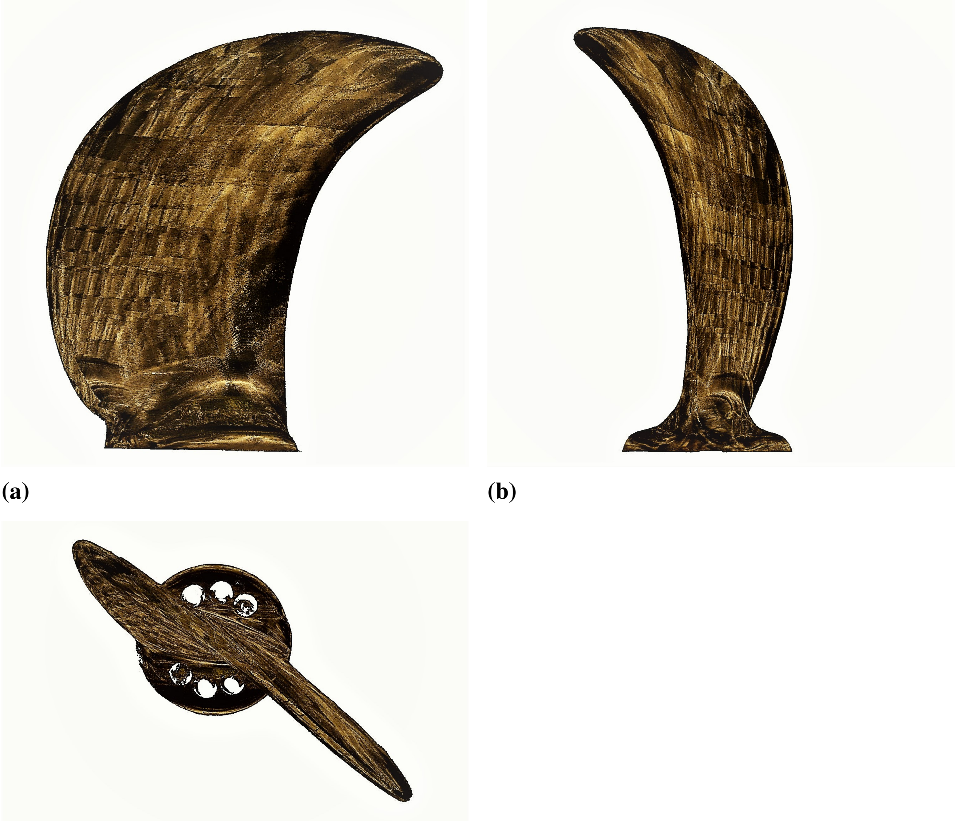
The scanning algorithm is terminated when a top edge and side edge is detected in the same image. This is a simple solution that worked in this particular project, but there might be some problems with this method for other types of propeller blades. If the propeller blade is quite small, a top edge and side edge might be found in the same image long before the scanner reaches the tip of the blade. For blades with very high skew, the edge on the underside of the blade tip could be classified as a top edge, and thus terminating the scanning process before the topside of the blade tip can be scanned.

The algorithm currently only scans one side of the blade at a time. To scan the other side of the blade the algorithm needs to be implemented again on that side. There is also a problem with getting good scans all around the edges. For this project, some scans of the edges was done by moving the robot to certain scan poses to capture parts of the geometry not included in the initial point cloud.

The original point cloud consisted of more than 500 million points, resulting in a large of file size of about 20 GB. Due to computational constraints, the point cloud was sampled down to about 4.5 million points to be able to work with the point cloud in MATLAB and other software. This sampling was done using a uniform voxel grid filter, removing points by the same factor over the whole point cloud. The problem with using this method, is that information is lost in areas where a higher degree of measurement accuracy and resolution might be wanted. An ideal solution for the sampling of points could in stead be weighted to include more points where we want more data, for example around the edges of the blade.

The completed point cloud of the propeller blade is seen from the front, the side and the top in figure 4.5.

Figure 4.5 The combined point cloud from the scanning process of the blade, which is to be used for the further estimation of design parameters. Some noise has been segmented out manually. Seen from the front (a), side (b), and top (c).



(a)

(b)

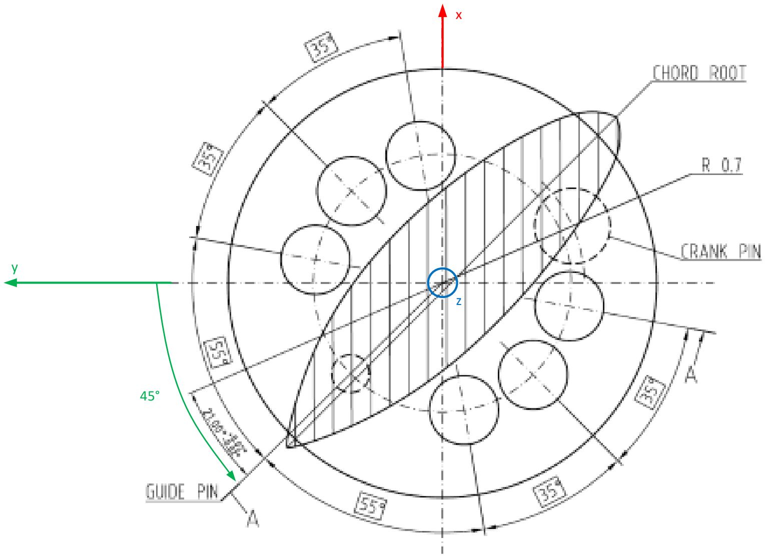
(c)

4.4 Defining coordinate system

The first thing that needs to be done to accurately find the design parameters of a propeller, is to define the correct origin and orientation of the coordinate system. We first set the origin of the coordinate system as the point where the estimated blade center axis intersects the bottom plane of the blade foot. The z-axis is set as vertically upwards through the reference line. All points on the propeller reference line is then defined with an x-value of 0 and an y-value of 0.

The hub diameter of the propeller is assumed to be known. This applies to both the in-line quality control scenario, where the hub diameter is given in the design propeller drawing, as well as in the reverse engineering scenario. For reverse engineering purposes, the propeller blade would in most cases be mounted on a hub on a ship already, and the hub diameter could be easily measured. It is hard to imagine a scenario where a single controllable pitch propeller blade is required to be reverse engineered, and the hub it will be attached to is not present or its dimensions are unknown. In our case, the distance from the origin of the coordinate system to the bottom of the blade foot is listed as 372 mm in the propeller drawing. This information is used as the translation of the point cloud in the z-direction, so that all points on the flat bottom plane of the blade foot are now defined with coordinates $(x, y, 372)$.

Figure 4.6 The figure shows a segment of the propeller drawing. The view is from the top, looking straight down on the blade. For the correct blade orientation, a guide pin on the underside of the blade is located with its center axis at an angle of $45°$ relative to the y-axis, which is positive towards port of the ship.



The location of the origin and the direction of the z-axis is important to know, but for controllable pitch propeller blades, the orientation of the blade needs to be correctly oriented at a specific angle relative to the x and y axis as well. The orientation of the blade refers to the blade's or point cloud's rotation about the z-axis. As stated in chapter 2, the parameters of controllable pitch propellers are defined with the propeller oriented in a specific way so that the pitch at 0.7R is equal to a given value, called the design pitch. However, finding the correct orientation of the point cloud by iteratively using the rotation matrix to rotate the whole point cloud, and calculate the pitch at 0.7R each time - for comparison with the design pitch - would be a very slow and computing intensive solution. Thus, in this project, the blade was simply oriented manually to the orientation providing the most accurate result for the pitch calculation at 0.7R.

The best solution for automation of the process of defining the coordinate system however, would be to first scan the underside of the blade. Then RANSAC could be used to find the cylinder representing the center hole of the blade foot. The center axis of this cylinder would represent the propeller reference line. Then, the center axis of the guide pin hole under the blade could be used to draw a line from the reference line to the center axis of this pin hole. For the particular propeller blade scanned in this project, this line should be oriented at an angle of $45°$ relative to the y-axis. This can be seen in figure 4.6, referencing the propeller drawing.

This method would provide a more accurate estimation of the z-axis, as well as a more precise orientation of the blade relative to the y-axis, than the method that was used in this project. Test scans of the underside of the blade were done after lifting the blade up, but there were problems with aligning these scans with the previously acquired point cloud of the propeller. Thus, the first estimation of the center of the whole blade foot based on MSAC became the solution for defining the z-axis in this project. We will keep this in mind when reviewing the results.

All these problems with defining and orienting the correct coordinate system, which is at a point below the point cloud itself, show why the reverse engineering of a controllable pitch propeller blade is much more difficult than the reverse engineering of a fixed pitch propeller. For the scanning of a propeller with all blades mounted on the hub, the origin of the coordinate system will be located within the point cloud itself if the hub is also scanned. The orientation of the blades relative to the hub are also fixed, so the problems with orientation are also much easier to overcome for a fixed pitch propeller.

The final orientation of the coordinate system can again be seen as the local frame (solid lines) in figure 2.3.

4.5 Extracting radial sections from the propeller point cloud

With the completed point cloud of the propeller blade aligned to the correct coordinate frame, as described previously in the chapter, this is the point where the work with extracting measurement data or geometric design parameters begins. From the theory in chapter 3, we see that the design parameters are given for radial sections of the propeller. The radii of these sections are defined as fractions of the total propeller radius.

The estimated propeller radius of the scanned propeller is found by measuring the distance from every point in the point cloud to the origin in the yz-plane. From this, the sections of particular interest at 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 0.95 of the measured radius are found. These are the radii at which the ISO standard for propeller measurements specify that the sections must be measured. The points of these sections are shown in Figure 4.7 as black dots in the point cloud of the propeller.

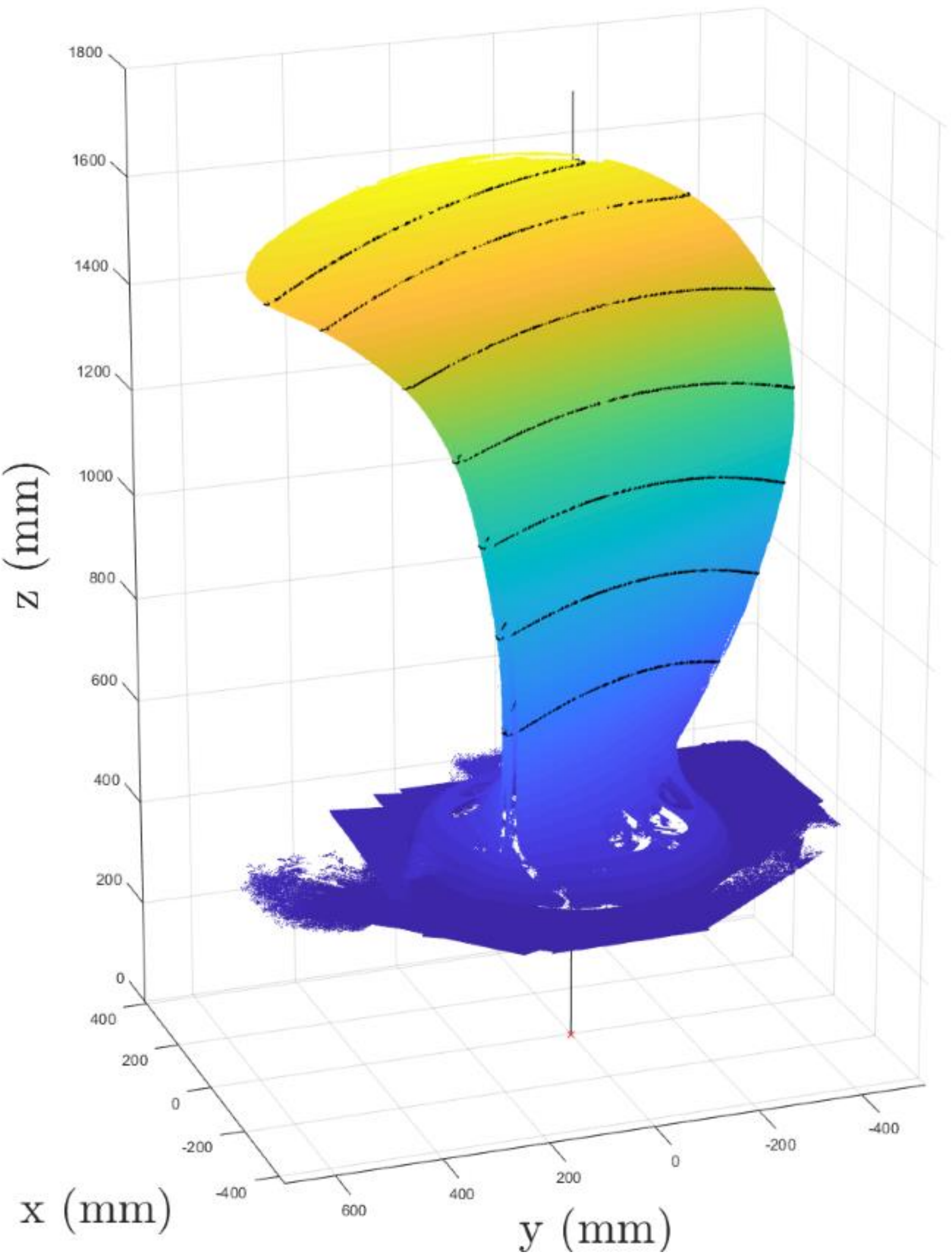
We want to extract these sections of interest from the point cloud for further analysis. An algorithm was implemented to segment out the points that are the correct distance away from the origin in the y and z directions. No points are at the exact distance away from the origin that is specified, so a small threshold is applied to include points that are very close to this distance. Algorithm 1 shows the pseudocode for this process.

Algorithm 1 The algorithm for segmenting out radial sections from a point cloud (Mx3 matrix of (x, y, z) points) of a propeller blade. Also, a list of radii $(radius)$ is needed to define the sections to be segmented out.

```
for i = 1:numberofsections do
    for j = 1:numberofpoints do
        if $abs\left(radius(i) - \sqrt{y(j)^2 + z(j)^2}\right) \leq threshold$ then
            $section(j,:) = [x(j), y(j), z(j)]$
        else
            $section(j,:) = 0$
        end if
    end for
    (Remove rows of zeros from the section, and continue with 'opening out' and
    estimation of the design parameters for the section)
end for
```

These points, represented as black dots in Figure 4.7, are then used to estimate the design parameters for each section. The same segmented out cylinder sections from the point cloud can be seen in Figure 4.8a.

Figure 4.7 Points of the point cloud that are found to lay at certain radial distances are shown as black dots. The sections are at 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95 of the total propeller radius.

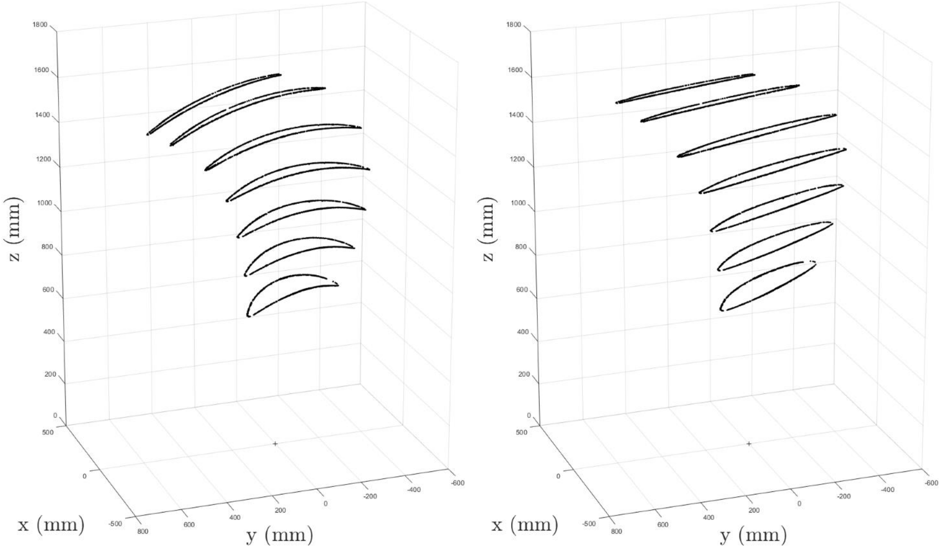


4.6 Opening out of cylinder sections

To be able to calculate the design parameters for each cylinder section of the propeller blade, the cylinders need to be ‘opened out’ as described in Chapter 2. The points are then laying along a plane parallel to the xy-plane, and with a distance from the xy-plane equal to the radial distance of the cylindrical sections from the origin.

The opening out of the cylinder section is done by setting the new z-coordinate for each point in the section equal to the radial distance to the original point (r), and the new y-coordinate equal to the arc length spanning from the z-axis to the point with a radius of r.

Figure 4.8 Opening out of segmented sections from the point cloud.



(a) Segmented sections at predefined radii **(b)** Opened out sections

The specific formulas used to calculate the point coordinates in the opened out sections, will now be presented. If (x_1, y_1, z_1) are the coordinates of an original point in the cylinder section, the radius r is

$$r = \sqrt{y_1^2 + z_1^2} \tag{4.2}$$

and the skew angle of the point, (ϕ), is

$$\phi = \cos^{-1}\left(\frac{z_1}{r}\right) \tag{4.3}$$

The coordinates (x_2, y_2, z_2) of the flattened out section are then found by:

$$x_2 = x_1 \tag{4.4}$$

$$y_2 = r\phi \,\mathrm{sgn}(y_1) \tag{4.5}$$

$$z_2 = r \tag{4.6}$$

The result of this process can be seen in Figure 4.8b. The figure shows the same cylinder sections as in Figure 4.8a, but now the cylinders have been opened out to allow easier calculations of the propeller design parameters. We will now look into the specific process of calculating the design parameters of these opened out sections.

4.7 Calculating design parameters for each section

Continuing, we will be looking at the section at 0.7R to see how the design parameters of the section are found. This section can be seen as the most important section to measure on propeller blades, since this is where the design pitch of the propeller is defined. We will check whether the pitch of this section corresponds with the design pitch later on.

Figure 4.9 The opened out section of points at 0.7R, shown in the xy-plane. The propeller reference line has the coordinates (0,0).

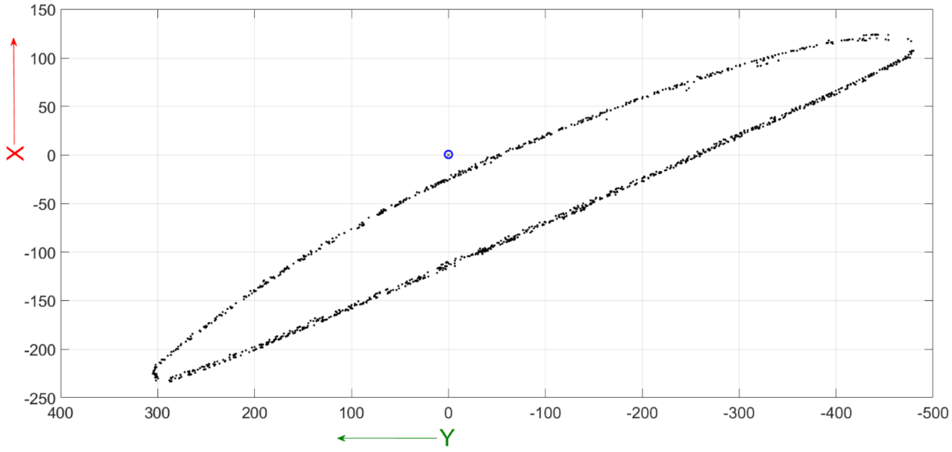


Figure 4.9 shows the opened out section as viewed in the xy-plane, similar to the view of a propeller section in Figure 2.7. We note that there are some measurement inaccuracies in the section of points. The scans from each side of the blade also don't seem to be aligned perfectly. The causes and consequences of these issues will be further discussed in chapters 5 and 6.

From here, the first thing that needs to be done, is to estimate the trailing and leading edges of the section. The way this was done in this project was to find the centroid of the points in the section, shown as a green cross in Figure 4.10. The centroid is simply the mean position of all the points in the section. From there the distance from each point in the section to the centroid are calculated, and the points with the largest distance from the centroid on each side along the y-axis were determined to be the trailing and leading edge of the section.

This solution for finding the edge points worked relatively well in this particular project, but other methods of doing this could be examined to find a more robust and accurate solution. One interesting method to investigate would be to model the leading and trailing edge in the point cloud using B-spline surfaces, and finding the points where the radial sections intersect the B-splines. This type of design model for a propeller blade is presented in [28].

Figure 4.10 Using the centroid of the points in the section, the trailing and leading edges are determined as the points furthest away from the centroid, and with a greater or smaller y value than the centroid, respectively.

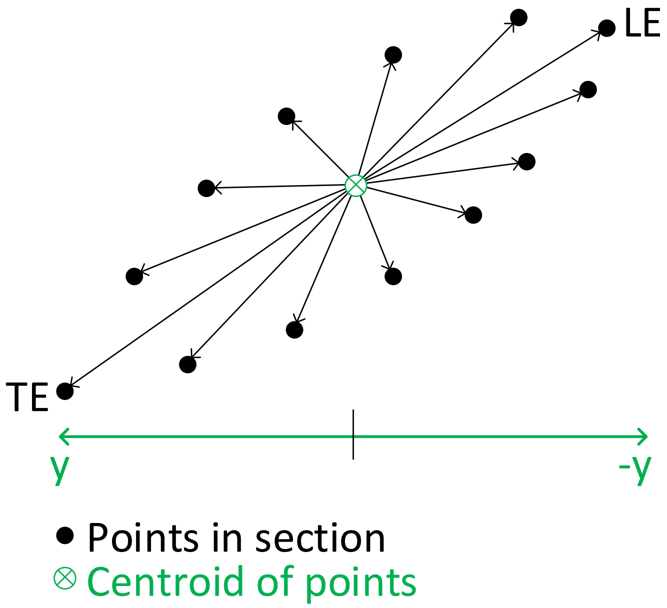
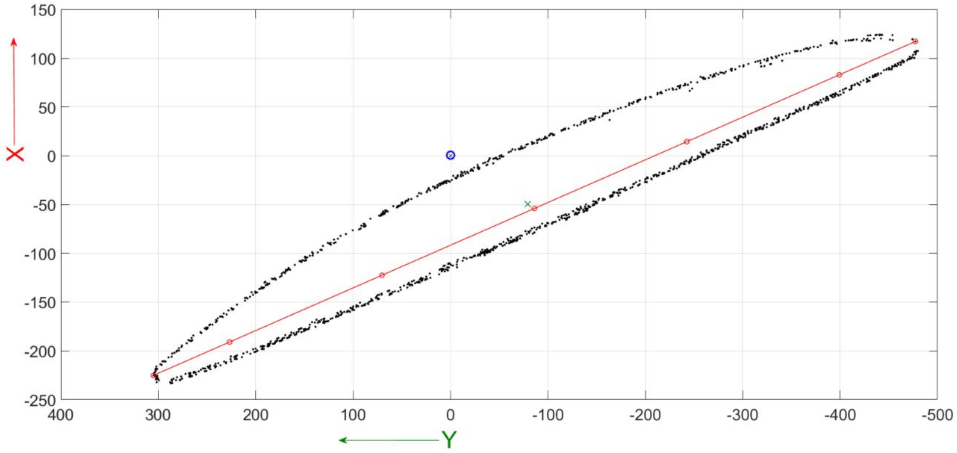


Figure 4.11 The chord line from the trailing edge to the leading edge is added to the section with points for thickness measurements added. The centroid of the section of points is show as a green X.



When the trailing and leading edges are found, the chord line is easily plotted as the direct line from the trailing to the leading edge. At points along the chord line the thickness of the section should be measured, with the point at 50% of the chord line, also known as the mid-chord, being important for finding the skew and rake of the section. The chord line with measurement points at $(10\%, 30\%, 50\%, 70\%, 90\%)$ of the chord is shown in figure 4.11. We can then look at the specific methods used for calculating each of the propeller design parameters.
The pitch angle (defined in the propeller drawing as α, not θ as found in the literature) of the section is determined by

$$\alpha = \tan^{-1}\left(\frac{x_{LE} - x_{TE}}{y_{LE} - y_{TE}}\right), \tag{4.7}$$

where (x_{LE}, y_{LE}) are the x and y coordinates for the leading edge and (x_{TE}, y_{TE}) are the coordinates for the trailing edge. Using equation 2.1, the pitch distance is then found by

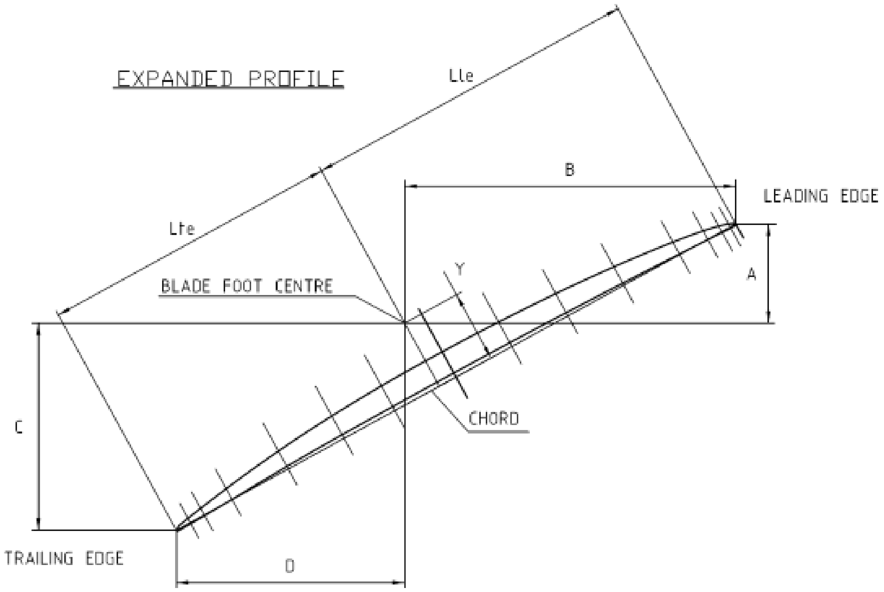
$$P = \tan(\alpha)2\pi r, \tag{4.8}$$

where r is the radius of the cylinder section. The chord length (L) is

$$L = \sqrt{(x_{LE} - x_{TE})^2 + (y_{LE} - y_{TE})^2} \tag{4.9}$$

We want to compare our values of the design parameters with the values provided in the propeller drawing.

Figure 4.12 A segment from the propeller drawing showing an opened out section of the blade, and how the rest of the design parameters are defined for this section is defined.



If we look at the expanded profile in the propeller drawing shown in figure 4.12, we see how the rest of the parameters for a section are defined. The blade foot centre is the propeller reference line, and so it's x and y coordinates are $(0, 0)$. This particular propeller designer uses slightly different dimensions than what is outlined in chapter 2, and so we want to calculate these dimensions instead of using the formulas described earlier in this report. The distances defining the edge positions are found by

$$A = x_{LE},\ B = y_{LE},\ C = -x_{TE},\ D = -y_{TE} \tag{4.10}$$

The two lengths Lle and Lte define the skew of the section, and are found by

$$Lle = \sin(\beta)\sqrt{A^2 + B^2} \tag{4.11}$$

$$Lte = L - Lle \tag{4.12}$$

where the angle β in radians is

$$\beta = \text{atan2}(A, B) + \frac{\pi}{2} - \alpha \tag{4.13}$$

The last parameter, Y, represents the rake of the section. It is found by

$$Y = \frac{Lle}{\tan(\beta)} \tag{4.14}$$

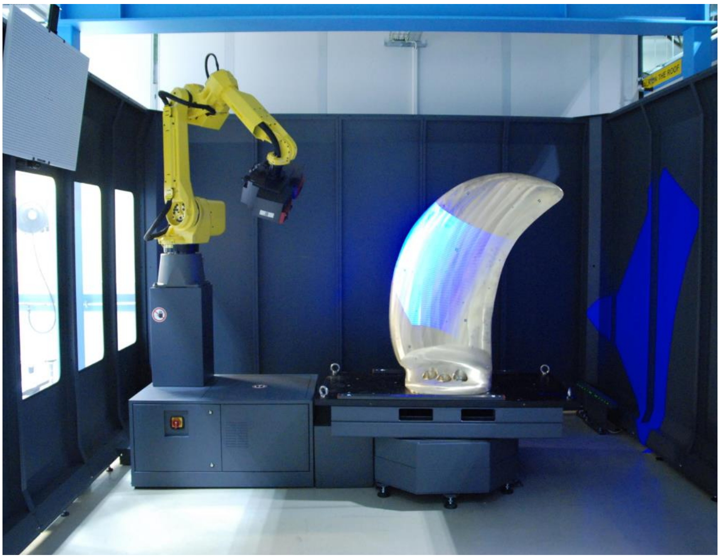
In the next chapter, the results from the calculations of these design parameters for each section of the blade will be presented.

Chapter 5

Results

In this chapter, we will look at the results from the process of estimating the design parameters of the propeller. Two different scans were made of the propeller blade in this project. The first is the scanning procedure described in Chapter 4. The other scan was done with an ATOS ScanBox located at GKN Aerospace in Kongsberg, which is a system with very high accuracy. We will compare the quality of the two scans later in this chapter. Figure 5.1 shows the ATOS scanner in action scanning the blade.

Figure 5.1 The propeller blade being scanned with ATOS ScanBox.



This scanning system outputs an .stl file. The .stl file was converted to a .ply file and aligned and oriented to the correct coordinate frame to be able to analyze the results from this scan.

We will begin with a note on the general scan quality from the system presented in this report and from ATOS ScanBox.

Figure 5.2 Left: section at 0.4R produced by the scanning system presented in this report. Right: section at 0.4R produced by ATOS ScanBox.

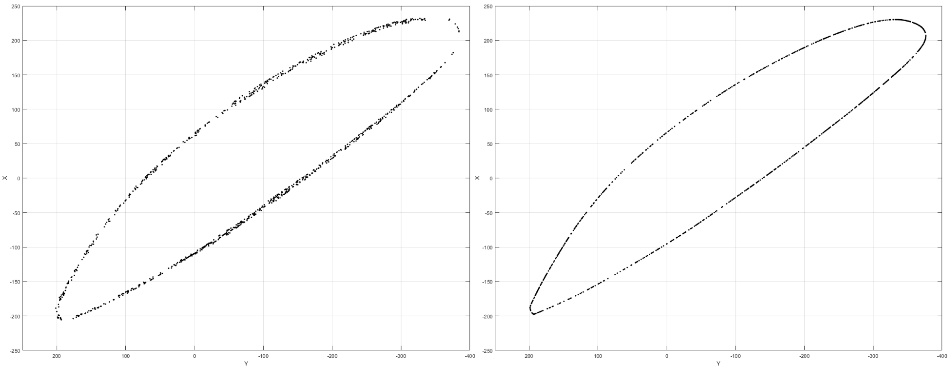


Figure 5.2 shows the blade section at 0.4R scanned with our system (left) and ATOS ScanBox (right). Two differences are particularly noticeable. In the section generated with the Zivid system, there are a lot of surface irregularities compared with the ATOS section, which has its points nicely aligned along the curves. The other difference is found in the alignment of the scans. Inaccurate alignments might be the cause of some of the surface irregularities in our system, in addition to noise caused by the reflectivity of the blade. But the most apparent misalignment is between the scanning procedure from each side of the blade. We see that the scan of the pressure side of the blade has been displaced relative to the suction side. The alignment of scans seem very impressive from the ATOS scan. We will look at the implications of the misalignment and surface irregularities from our system later in this chapter.

In the rest of this chapter, we will look at the specific results obtained from the two different scanning systems, and analyze any systematic or random errors produced by each of them.

5.1 Estimated design parameters with Zivid system

We will look at the results from the sections of the blade that the ISO standard specifies that must be measured. The results from the estimation of the design parameters of the propeller are shown in Table 5.1.

Table 5.1: Estimated design parameters from scan with Zivid.

r/R	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.95**
r	720.0	900.0	1080.0	1260.0	1440.0	1620.0	1710.0
P/D	0.931	0.950	0.960	0.963	0.988	0.923	0.878
P	3350	3420	3456	3465	3557	3323	3163
α	36.53	31.18	27.00	23.65	21.47	18.09	16.41
Lle	437.5	488.1	507.7	484.2	396.3	172.5	-63.3
Lte	280.6	284.4	314.0	369.7	479.0	646.8	743.7
Ltot	718.0	772.5	821.7	853.9	875.3	819.3	680.3
Y	45.5	62.0	74.2	84.1	99.9	110.4	100.9
A	223.8	199.7	164.4	117.2	52.1	-51.4	-114.7
B	378.6	449.7	486.1	477.2	405.4	198.2	-32.3
C	203.6	200.3	208.6	225.3	268.4	305.7	306.9
D	198.3	211.2	246.1	304.9	409.2	580.6	684.9

We will compare the values with the data given in the propeller drawing, and look at the deviation of each parameter from the design values.

5.1.1 Pitch

Table 5.2: Deviation between pitch values measured with Zivid 3D scanner (P_{Zivid}) and design values ($P_{drawing}$) of the sections.

r/R	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.95**
$\mathbf{P}_{Zivid}$	3350	3420	3456	3465	3557	3323	3163
$\mathbf{P}_{drawing}$	3276	3377	3424	3467	3460	3287	3071
$\mathbf{Error}_{mm}$	74	43	32	-2	97	36	92
$\mathbf{Error}_{\%}$	2.25%	1.29%	0.93%	-0.05%	2.82%	1.09%	2.99%

Mean	**StDev**
1.61%	1.03%

The first parameter we will look at is the pitch angle of the section. Table 5.2 shows the error or deviation of the pitch distance for each of the sections in percent, as well as the average error and the standard deviation of the measurements.

With the mean error of the pitch at 1.61% and standard deviation of 1.03%, there seems to be a slight trend that the pitch estimates produced from the scan with the Zivid camera have a consistent positive error compared to the design values. All of the values are larger than the design values except for the section at 0.7R, which is of course the value for which the point cloud has been oriented to match. The main reason for the errors in the pitch estimates seems to be the suboptimal alignment of the scans from each side of the blade, causing the leading edge to be moved in the positive direction along the x-axis. This results in the pitch angles measured becoming larger than the pitch angles from the propeller drawing. The fact that the pitch at 0.7 seems to not have this error compared to the other values, is mostly a coincidence. The section at 0.7 with chord line is shown in Figure 4.11. In this figure we can see that the misalignment of the pressure side relative to the suction side is displaced in the approximate direction of the chord line itself. Notice the gap close to the trailing edge and what seems to be two overlapping surfaces near the leading edge. This displacement along the chord line results in the pitch angle at 0.7 not changing much due to the misalignment.

5.1.2 Chord length

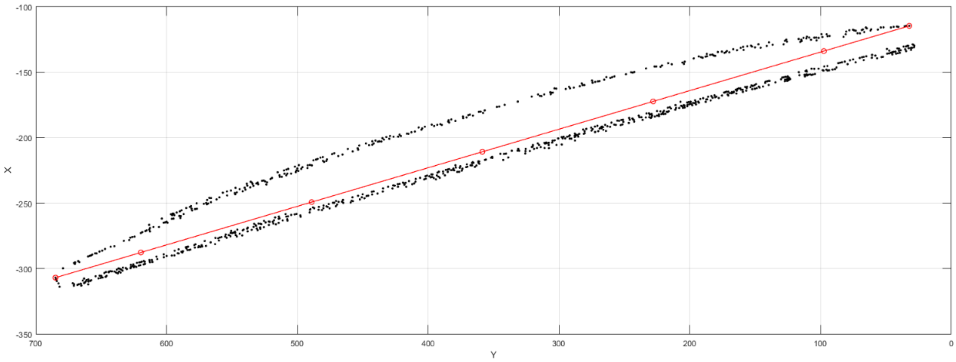
The deviation of the measured chord lengths from the design values are shown in Table 5.3.

Table 5.3: Deviation of chord lengths measured with Zivid camera and design values.

r/R	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.95**	**Mean**	**StDev**
Error$_{mm}$ Ltot	18.1	15.0	17.3	15.1	20.3	19.2	-3.3	14.5	7.5
Error$_{\%}$ Ltot	2.6%	2.0%	2.2%	1.8%	2.4%	2.4%	-0.5%	1.83%	0.97%

There is one clear outlier in the errors, at the 0.95R section. For the other sections, up to 0.9R, there is a mean error of 2.22% and a standard deviation of 0.27%. This shows a clear and consistent positive error in the measurement of the chord lengths. This can be easily explained by the misalignment of the scans of the two sides of the blade, causing the chord length measured to be longer. The 0.95R section is interesting to look at since it significantly differs from the other values. This section is shown in Figure 5.3.

Figure 5.3 The 0.95R section missing a part of the leading edge.



The reason for the difference becomes apparent. The leading edge of the 0.95R section is located near the very top of the blade. During the scanning there were problems with getting scans of the top edge due to reach of the robot. Thus, there are missing points for the leading edge in the point cloud. The chord length at 0.95R is measured from the trailing edge to a point closer than what would have been the leading edge point.

5.1.3 Skew and rake

We are now interested in the parameters defining the skew and rake of the section. These are the parameters Lle and Lte, determining how much the chord line is skewed in relation to the blade center, and Y, determining the distance from the blade center to the chord line (rake). When looking at the parameters defined in relation to the origin of the coordinate system it doesn't make sense to look at the errors as percentages. If the design value for a parameter is 0.1mm from the origin and it is measured as 1mm, the error is really quite small, but will show up as an error of 900% if measured in percent. Therefore, for these parameters we look at the deviation from the design value in millimeter. This is shown in Table 5.4.

Table 5.4: Deviation of skew and rake parameters measured with Zivid camera and the design values.

r/R	0.4	0.5	0.6	0.7	0.8	0.9	0.95	Mean	StDev
Error$_{mm}$ Lle	3.0	4.6	7.3	7.8	13.2	17.4	-3.8	7.1	6.4
Error$_{mm}$ Lte	15.2	10.4	10.0	7.3	7.1	1.7	0.6	7.5	4.7
Error$_{mm}$ Y	2.7	9.5	12.1	11.2	16.3	18.3	7.6	11.1	4.9

The mean error values of the two distances Lle and Lte are obviously affected by the mean chord length being estimated to be larger than the real value, since they are defined along the chord line. The means of 7.1mm and 7.5mm are expected as they add up to the total chord length mean error of 14.5mm. However, looking at the errors, an interesting pattern is observed. The deviations do not appear to be randomly distributed around the mean. Instead, the distance to the leading edge (Lle) appears to be increasing relative to the design distance further up on the blade. Similarly, the relative distance to the trailing edge is decreasing. The value at 0.95R can be disregarded as the leading edge point is clearly wrong. This phenomenon will be further discussed later in this chapter.

The deviation of the Y value, or the rake of the section, also seems to be following a similar pattern, increasing further up on the blade. For this parameter, this seems to be happening due to a combination of two factors: the blade center axis not being located precisely enough, and the misalignment of the scans causing an increasing error as the pitch angle decreases. However, this should be investigated further to determine the root causes.

5.1.4 Edge locations

The last few parameters we will look at is the parameters A, B, C and D, defining the locations of the leading and trailing edge. The errors in millimeter of these parameters for the Zivid scan can be seen in Table 5.5.

Table 5.5: Deviation of edge locations measured with Zivid camera and design values.

r/R	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.95**	**Mean**	**StDev**
Error$_{mm}$ A	3.6	-3.1	-5.5	-7.1	-6.7	-11.4	-8.6	-5.55	4.44
Error$_{mm}$ B	1.6	7.7	11.4	11.6	17.7	22.3	-0.7	10.23	7.59
Error$_{mm}$ C	13.3	14.8	16.2	13.1	21.7	19.8	13.0	15.99	3.23
Error$_{mm}$ D	8.4	2.9	2.7	2.2	-1.7	-5.0	-4.0	0.80	4.34

The errors for these parameters are obviously affected by the misalignment of the scans. So some deviation from the design values are expected and observed, especially for the B and C parameters. There is also some trends in the errors of these parameters, where the error of A and D decreases, and the error of B and C increase further up on the blade. This can be explained as a skewing of the results as the pitch changes, due to a misplaced blade center axis.

5.2 Estimated design parameters from ATOS ScanBox

The method of estimating the propeller design parameters developed in this project was tested on the .ply-file produced from the ATOS system. The results from this process are shown in Table 5.6.

Table 5.6: Estimated design parameters from scan with ATOS ScanBox.

r/R	0.4	0.5	0.6	0.7	0.8	0.9	0.95
r	720.0	900.0	1080.0	1260.0	1440.0	1620.0	1710.0
P/D	0.906	0.938	0.947	0.963	0.939	0.880	0.830
P	3261	3376	3408	3468	3382	3167	2987
α	35.79	30.84	26.66	23.66	20.49	17.29	15.54
Lle	429.0	479.7	497.8	475.1	387.5	161.4	-38.2
Lte	273.3	277.9	308.3	363.7	471.5	639.9	738.5
Ltot	702.3	757.7	806.1	838.8	859.0	801.3	700.3
Y	43.9	57.3	66.6	76.1	82.8	92.1	91.7
A	215.2	196.7	163.9	121.0	58.1	-40.0	-98.6
B	373.7	441.3	474.8	465.7	392.0	181.5	-12.3
C	195.5	191.7	197.9	215.6	242.7	278.1	286.2
D	196.0	209.3	245.6	302.6	412.7	583.7	687.0

We will again look at the specific parameters and try to find any systemic errors or interesting results.

5.2.1 Pitch

We look at pitch first, comparing the pitch of each section with the design values. Table 5.7 shows the error or deviation of the pitch distance for each of the sections in percent, as well as the average error and the standard deviation of the measurements.

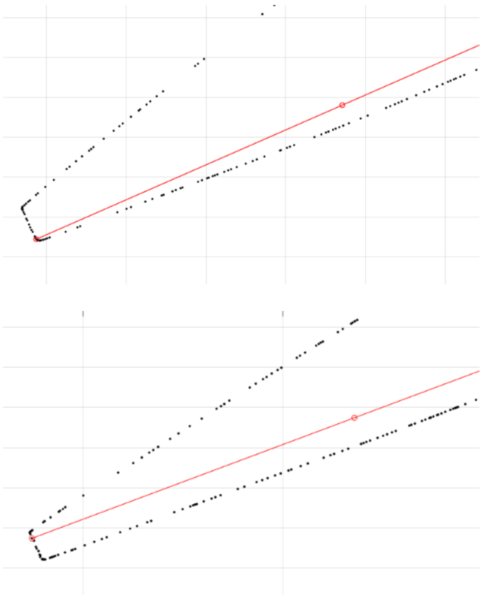
Table 5.7: Deviation between pitch values measured with ATOS ScanBox (P$_{ATOS}$) and design values (P$_{drawing}$) of the sections.

r/R	0.4	0.5	0.6	0.7	0.8	0.9	0.95
P$_{ATOS}$	3261	3376	3408	3468	3382	3167	2987
P$_{drawing}$	3276	3377	3424	3467	3460	3287	3071
Err$_{mm}$	-15	-1	-17	1	-78	-120	-84
Err$_{\%}$	-0.46%	-0.04%	-0.48%	0.02%	-2.26%	-3.64%	-2.72%

Mean	StDev
-1.37%	1.37%

Looking at the mean pitch error and standard deviation from the ATOS scan, it doesn't look like the results from this scan are significantly better for estimating pitch than the results from the Zivid scan. However, looking at the error for each section, they are clearly divided into two distinct groups. The first one, from 0.4R to 0.7R, produce quite accurate results with a mean error of only -0.24% and a standard deviation of 0.23%. The other group, from 0.8R and up, has a relatively consistent error with a mean of -2.87% and a standard deviation of 0.57%. The reason for this is found in the way the trailing edge of the blade is defined. With the method used in this project, the trailing edge point is found in the bottom corner of the flat trailing edge in the sections from 0.4 to 0.7. From 0.8R the trailing edge point is found in the upper corner of the edge. This difference is shown in Figure 5.4.

Figure 5.4 The trailing edges of sections are not consistently defined. Up to 0.7R, a point near one corner of the edge is determined as TE. From 0.8R, the TE is found near the other corner. Top: Trailing edge at 0.7R. Bottom: Trailing edge at 0.8R.



In reality however, the trailing edge point should be located in the middle of the flat edge. Since the point cloud is aligned to the correct pitch at 0.7R, the sections where the chord line is defined similarly will have similar pitch angles and therefore the error is small for these sections. When the trailing edge point is found in the upper corner of the edge, the pitch angle becomes smaller, so these sections have a consistent negative error compared to the design pitch.

5.2.2 Chord length

The error of the chord length values measured with ATOS ScanBox compared with the design values are shown in Table 5.8.

The chord length measurements are actually very close to the design chord lengths for the sections from 0.4 to 0.9R, with a mean error of 0.20% and a standard deviation of 0.17%. One result that clearly stands out is the error of 2.4% for the 0.95R section.

Table 5.8: Deviation of chord lengths measured with ATOS ScanBox and design values.

r/R	0.4	0.5	0.6	0.7	0.8	0.9	0.95	Mean	StDev
Error$_{mm}$ Ltot	2.4	0.2	1.7	0.0	4.0	1.2	16.7	3.7	5.5
Error$_{\%}$ Ltot	0.3%	0.0%	0.2%	0.0%	0.5%	0.1%	2.4%	0.52%	0.80%

The reason for this outlying error could perhaps be understood when looking at the other design parameters for the section.

5.2.3 Skew and rake

The deviations of the skew and rake parameters are shown in Table 5.9.

Table 5.9: Deviation of skew and rake parameters measured with ATOS camera and the design values.

r/R	0.4	0.5	0.6	0.7	0.8	0.9	0.95	Mean	StDev
Error$_{mm}$ Lle	-5.6	-3.8	-2.6	-1.3	4.4	6.3	21.3	2.7	8.6
Error$_{mm}$ Lte	7.9	3.9	4.3	1.3	-0.4	-5.2	-4.6	1.0	4.5
Error$_{mm}$ Y	1.1	4.8	4.5	3.2	-0.8	0.0	-1.6	1.6	2.4

The mean values of these parameters are much closer to the design values than the means from the Zivid scan. This is expected, since the alignment of the scans are much better from the ATOS scan, and thus the chord line is more accurately defined, as shown by the accurate chord length measurements. However, we still see the same pattern from Chapter 5.1 of the error of Lle increasing, and the error of Lte decreasing, further up on the blade. The error of Lle deviates from about -6 mm to about 6 mm around the mean for the sections from 0.4R to 0.9R. This is similar to the pattern observed for the same parameters from the Zivid scan. The reason for this is not exactly known, but since the same pattern is observed in both scans, there could be a possibility that the actual propeller blade deviates slightly from the design values. The edges of the blade is the part where there is least control over the geometry during manufacturing. Templates are used as the basis for finishing the edges during production, and there is little precise measurements of the blade edges.

The results from parameter Y are much closer to the design values than in the Zivid scan. This seems to be because the center axis was more accurately placed in the ATOS scan. The ScanBox system included a very precise part localization table, where the blade was precisely clamped to the correct position to allow the robot to know the exact placement of the center axis.

The 0.95R section has a much larger error for Lle than the other sections, and we can look at the errors for the edge locations to analyze this.

5.2.4 Edge locations

The error of the ATOS ScanBox measurements of the parameters defining the edges of the blade is shown in Table 5.10.

Table 5.10: Deviation of edge locations measured with ATOS ScanBox and design values.

r/R	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**0.95**	**Mean**	**StDev**
Error$_{mm}$ A	-5.0	-6.1	-6.1	-3.3	-0.7	0.0	7.5	-2.0	4.5
Error$_{mm}$ B	-3.3	-0.7	0.1	0.1	4.3	5.6	19.3	3.6	7.0
Error$_{mm}$ C	5.2	6.2	5.5	3.4	-4.0	-7.8	-7.7	0.1	5.9
Error$_{mm}$ D	6.1	1.0	2.2	-0.1	1.8	-2.0	-1.9	1.0	2.6

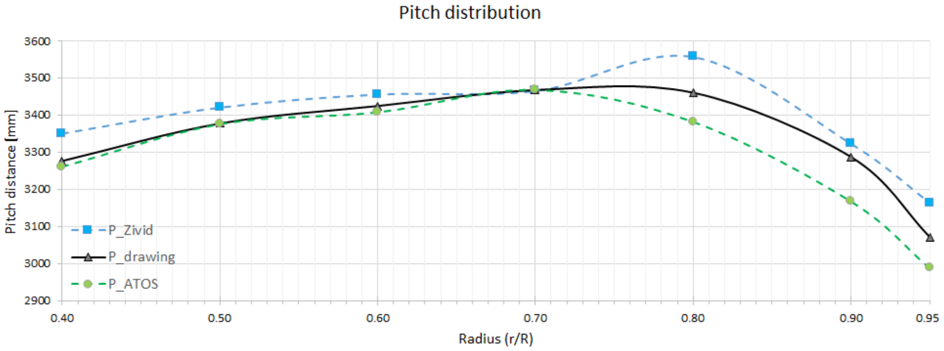
We see the same pattern from section 5.2.3, where the error of the parameters defining the leading edge (A and B) increase further up on the blade. The error for the trailing edge parameters (C and D) decrease. However, there is a difference from the results measured with the Zivid system, where the trend seemed to be ‘skewed’, so that the error for B increased while the error for A decreased. The results from the ATOS scan seem more like what would be expected from a system where the center axis is positioned correctly. The errors of the parameters for each of the edges is changing in the same direction when the pitch changes. The errors for the Y parameter also seem to indicate that the center point is better placed in the ATOS scan than in the Zivid scan.

The error of the B parameter in the 0.95 section is a much larger deviation than the error for the other parameters. This also causes the Lle of the section to be larger than expected. The 0.95 section is located close to the top of the blade, and due to the skew of the blade the leading edge part of this section is almost laying flat along the top edge of the blade. This means that very small deviations in the top edge from the design geometry could cause the length of the leading edge to be drastically changed.

5.3 Comparison

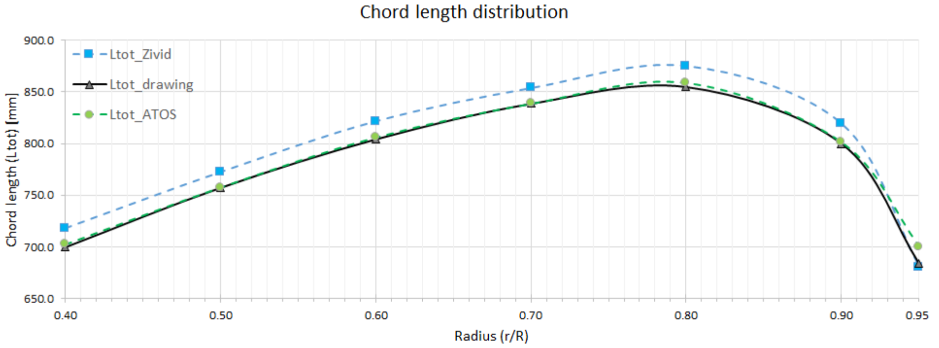
In this section, we will summarize the results from the optical metrology system developed in this project, and the state-of-the-art system ATOS ScanBox. Figure 5.5 shows the design pitch distribution, and the distributions measured with the two different systems.

Figure 5.5 The pitch distribution given in the propeller drawing, compared to the pitch distributions measured with Zivid and ATOS ScanBox.



The pitch estimated from the ATOS ScanBox seems very accurate for sections where the trailing edge is similarly defined. A better method of automatically finding the edge points for each section would likely increase the accuracy of the pitch measurements. The results of the pitch estimation with the Zivid system seem less consistent. There could be several reasons for this. First of all, the misalignment of scans introduces some errors in the edge positions. There is also considerable surface inaccuracies with the Zivid scanner, meaning that the system could find a point with the wrong position as one of the edges. Overall, the pitch estimation results do not seem terrible for a first test of each of the scanning systems.

Figure 5.6 The chord length distribution given in the propeller drawing, compared to the chord lengths measured with Zivid and ATOS ScanBox.



In Figure 5.6 the estimated chord length distributions are compared with the chord lengths from the propeller drawing. The chord length estimated from the ATOS scan are very accurate, except for one of the sections. The chord lengths measured with the Zivid system have a consistent error caused by the misalignment of scans of each side of the blade.

Table 5.11 shows a summary of the mean errors and standard deviations of the parameters estimated with each of the scanning systems tested.

Table 5.11: Comparison of mean error and standard deviation for the parameters estimated with Zivid and ATOS ScanBox.

	Zivid		**ATOS ScanBox**	
Parameter	**Mean**	**StDev**	**Mean**	**StDev**
P	1.61%	1.03%	-1.37 %	1.37%
α	1.46%	0.94%	-1.26%	1.30%
Ltot	1.83%	0.97%	0.52%	0.82%
Lle	7.1 [mm]	6.4 [mm]	2.7 [mm]	8.6 [mm]
Lte	7.5 [mm]	4.7 [mm]	1.0 [mm]	4.5 [mm]
Y	11.1 [mm]	4.9 [mm]	1.6 [mm]	2.4 [mm]
A	-5.5 [mm]	4.4 [mm]	-2.0 [mm]	4.5 [mm]
B	10.2 [mm]	7.6 [mm]	3.6 [mm]	7.0 [mm]
C	16.0[mm]	3.2 [mm]	0.1 [mm]	5.9 [mm]
D	0.8 [mm]	4.3 [mm]	1.0 [mm]	2.6 [mm]

The standard deviation of the results seem quite similar in the two systems. However, the mean errors from the ATOS system are much closer to the design values. This indicates that some of the systematic errors outlined from the Zivid scan are not present in the scan from ATOS. These systematic errors will be further discussed in Chapter 5.4. In conclusion, the ATOS system is currently a more accurate 3D metrology system for estimating propeller design parameters than the system developed in this project. But the results from our system are encouraging, considering that there are clear and apparent ways the system can be improved to increase the accuracy.

5.4 Sources of error

Some of the sources of error that could cause inaccuracies in the results from the developed scanning system is as follows.

- **Misplaced center axis and orientation of coordinate system.**

  The method of using RANSAC for detecting the sphere shape on the blade foot to estimate the blade center axis is not accurate enough. This also seems like the reason for some of the misalignment between the scans from the two sides, as the estimated center axis is used to combine the point clouds. There is also a problem with getting the correct orientation of the point cloud relative to the origin, and currently this is done manually.

- **Inaccuracy in the camera calibration.**

  The camera calibration might not be perfect, and could account for some errors with regards to the alignment of scans.

- **Inaccuracies in the robot and pose estimation.**

  There could be small errors in the robot calibration and its kinematic parameters. This could result in small errors in the pose estimation of the end effector.

- **Inaccuracies in the 3D scanner.**

  There could be some inaccuracies in the 3D scanner's internal calibration as well.

- **Measurement errors due to reflections from the blade and noise from surrounding light sources.**

  The reflectivity of the surface looks to be causing some errors and 'peaks' on the surfaces of the point cloud. This is particularly a problem with regards to precise thickness measurements of the blade.

- **Sampling of points is not weighted to include more points where the geometry is more interesting.**

  The uniform voxel grid used for sampling down the point cloud removes points equally across the whole point cloud. Because of this, some inaccuracy can occur in the places where there is more need for measurement data than the rest of the blade, like the edges of the blade.

- **The method of finding trailing and leading edge is not optimal.**

  The method used for defining the edges of the section by calculating the point distance from the centroid clearly produces some bad results, with the edges sometimes being located several millimeters away from the real edge point.

- **Segmenting out too many points.**

  In the segmentation of points for the blade sections, a threshold of about ± 0.2 mm around the radius of the sections was used to get enough points for analysis. This could cause some inaccuracies in the point positions when the sections are opened out.

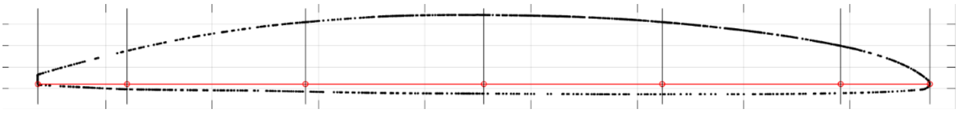
Chapter 6

Conclusion

6.1 Discussion

In this report, we have focused on finding the design parameters defined by the chord line of the sections, and not on measuring the thickness or camber. Some methods of measuring the thickness from the point clouds have been investigated, but no solution has proven to be as robust and accurate as hoped. Of course, the thickness is something that we ideally would like to measure, as this is a crucial part of the blade geometry. One idea could be to use the rotation matrix and rotate the sections by their pitch angle so the chord line is laying flat. This is shown in Figure 6.1.

Figure 6.1 Measuring the thickness for points along the chord line of a section.



If we draw a line straight up through the points on the chord line where we want to measure the thickness, the points closest to the line could be used to find the thickness. The problem however, is that we do not know which points belong to the suction side and which points belong to the pressure side of the blade. And the chord line is not always located in between the two blade sides. So perhaps a method based on first finding the point closest to the line could be used. This would be the point with the closest x value to the point on the chord line. Then, the next closest point could be found, but now looking for the closest point outside of a certain range of y values around the first point. In this way, a point on both the suction and pressure side can be found that can be used to estimate the thickness at that part of the section.

The two different problems described in Chapter 1 was reverse engineering and in-line quality control of a propeller blade. For both of these problems, the thickness measurements of the sections are also needed. In the case of reverse engineering, the estimated parameters and the thickness measurements should be enough to create a complete CAD model of the scanned propeller, as this is what is currently used today.

For in-line quality control, the system presented looks promising, if a method of finding the thicknesses is implemented. For the application of robotic grinding, a robot is needed anyways. So the scanning and grinding could potentially be done by the same robot, changing its end effector automatically to perform both tasks. This would be a much cheaper solution than using the ATOS ScanBox, which has a price of several million NOK. It could also prove to be a better solution for complete automation of the blade manufacturing process. With a system like ATOS, the blade would have to be moved between the scanning system and the grinding robot. The ScanBox also requires markers to be placed in the scanner's view.

In a setup with the same robot doing the scanning and grinding, the process would be both faster and require less human resources. In addition, the process could more easily be done iteratively for higher accuracy. Since the blade will be grinded anyways, there could be a possibility of spraying it with a matte paint to reduce the scanning errors due to reflectivity. An optimal solution, however, would avoid this as it likely would have to be sprayed in a different location than the robot with the 3D scanner.

6.2 Recommendations for future work

To improve the accuracy of the system, there are several things that could be done. First of all, the underside of the blade should be scanned as described in Chapter 4.4, to more accurately find the blade center axis and the propeller's orientation.

A method of measuring the section thicknesses should be found, either based on the idea from Chapter 6.1, or by finding another method. There should also be developed a new method of finding the leading and trailing edges of the sections, as this process was not perfect in this project.

Better calibration of the system and alignment of scans could be achieved by using an external position measurement system with the position and rotation of a high accuracy probe as its output. If the probe is identified in the scans, this could be used to accurately align scans.

In this project, the Zivid 3D scanner was set up fairly quickly, to be able to scan the blade in time. There are a lot of settings for the scanner that could be experimented with to better cope with the reflective bronze surface of the blade.

A new and smarter way of sampling the point cloud down to fewer points should be found. The goal should be to keep points where the geometry is more interesting and where the accuracy of the measurements are the most important. Points where there is nothing interesting to measure should be discarded.

Finally, the system should be tested on propeller blades with a wide variety of geometries, to get closer to a system that could be used industrially for all types of controllable pitch propellers.

6.3 Final conclusion

A system was presented that can automatically 3D scan a controllable pitch propeller blade. A method of estimating the design parameters of the propeller from the resulting point cloud of the 3D scan was outlined. The system could potentially be used for both reverse engineering and in-line quality control of the blades. Reverse engineering of controllable pitch propeller blades is a more challenging problem than reverse engineering of fixed pitch propellers. Some ways to overcome these challenges have been presented.

The initial results were promising, considering this was a first test of the system and that there are clear and obvious ways of improving it. However, there are some systematic errors in the results, and these need to be corrected before the system can be used in an industrial setting. If improvements are made on the system, it could prove to be a much cheaper solution for optical in-line quality control of propeller blades than the current state-of-the-art systems.

The system is designed with the purpose of it possibly being used in an automated grinding process for finishing propeller blades after they have been cast. If this sort of system is implemented successfully it could dramatically reduce the costs of propeller manufacturing.

Bibliography

[1] 3D Systems, Inc., 2017. Sense 3D scanner Tech specs. `https://www.3dsystems.com/shop/sense/techspecs`, accessed: 2018-06-06.

[2] Abbott, I. H., Von Doenhoff, A. E., 1959. Theory of wing sections, including a summary of airfoil data. Courier Corporation.

[3] Andersen, M. R., Jensen, T., Lisouski, P., Mortensen, A. K., Hansen, M. K., Gregersen, T., Ahrendt, P., 2012. Kinect depth sensor evaluation for computer vision applications. Technical Report Electronics and Computer Engineering 1 (6).

[4] Asimakopoulos, O. A. A., Kaklis, P., 2016. Effects of propeller geometry on cavitation. University of Strathclyde.

[5] Babicz, J., 2015. Wärtsilä Encyclopedia of Ship Technology - Second edition. Wärtsilä Corporation.

[6] Besl, P. J., McKay, N. D., 1992. Method for registration of 3-d shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures. Vol. 1611. International Society for Optics and Photonics, pp. 586–607.

[7] Bouguet, J.-Y., 2002. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`, accessed: 2018-06-12.

[8] Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[9] Cai, C., Dean-León, E., Mendoza, D., Somani, N., Knoll, A., 2013. Uncalibrated 3d stereo image-based dynamic visual servoing for robot manipulators. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, pp. 63–70.

[10] Callieri, M., Fasano, A., Impoco, G., Cignoni, P., Scopigno, R., Parrini, G., Biagini, G., 2004. Roboscan: an automatic system for accurate and unattended 3d scanning. In: 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on. IEEE, pp. 805–812.

[11] Canny, J., 1987. A computational approach to edge detection. In: Readings in Computer Vision. Elsevier, pp. 184–203.

[12] Carlton, J., 2007. Marine Propellers and Propulsion - Second Edition. Elsevier Science.

[13] Creaform Inc., 2017. Creaform HandySCAN 3D. https://www.creaform3d.com/sites/default/files/assets/brochures/files/handyscan3d_brochure_en_hq_21032017_2.pdf, accessed: 2018-06-06.

[14] D'Epagnier, L. K. P., Chung, H.-L., Stanway, M. J., Kimball, R. W., 2007. An open source parametric propeller design tool. In: OCEANS 2007. IEEE, pp. 1–8.

[15] Duda, R. O., Hart, P. E., 1972. Use of the hough transformation to detect lines and curves in pictures. Communications of the ACM 15 (1), 11–15.

[16] Egeland, O., 2017. A note on vision. Department of Mechanical and Industrial Engineering, NTNU Trondheim.

[17] Fischler, M. A., Bolles, R. C., 1987. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: Readings in computer vision. Elsevier, pp. 726–740.

[18] Foto von Stahlkocher, Wikimedia, 2005. A ship's controllable pitch propeller. https://commons.wikimedia.org/wiki/File:Verstellpropeller_eines_Hurtigrutenschiffes.jpg, accessed: 2018-05-22.

[19] Germanischer Lloyd, 2009. Materials for Propeller Fabrication. Germanischer Lloyd Aktiengesellschaft, Hamburg.

[20] GOM, 2018. ATOS ScanBox. https://www.gom.com/fileadmin/user_upload/systems/gom_atos-scanbox_EN.pdf, accessed: 2018-06-06.

[21] ISO 484-1:2015(E), 2015. Shipbuilding - Ship screw propellers - Manufacturing tolerances. Standard, International Organization for Standardization, Geneva, CH.

[22] ISO 484-2:2015(E), 2015. Shipbuilding - Ship screw propellers - Manufacturing tolerances. Standard, International Organization for Standardization, Geneva, CH.

[23] ITTC, 2014. International Towing Tank Conference - Dictionary of Hydromechanics - Propeller. ittcwiki.org, accessed: 2018-05-20.

[24] Larsson, S., Kjellander, J. A., 2008. Path planning for laser scanning with an industrial robot. Robotics and autonomous systems 56 (7), 615–624.

[25] Marine Insight, 2017. Parts of screw propeller. https://www.marineinsight.com/naval-architecture/controllable-pitch-propeller-cpp-vs-fixed-pitch-propeller-fpp/, accessed: 2018-05-21.

[26] Menna, F., Troisi, S., 2010. Low cost reverse engineering techniques for 3d modelling of propellers. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 38 (Part 5), 452–457.

[27] Oy, V. C., 2000–2018. Visual Components Premium 4.0: Proprietary. visualcomponents.com/.

[28] Pérez-Arribas, F., Pérez-Fernández, R., 2018. A b-spline design model for propeller blades. Advances in Engineering Software 118, 35–44.

[29] Ritter, J., 1990. An efficient bounding sphere. Graphics gems 1, 301–303.

[30] Sansoni, G., Trebeschi, M., Docchio, F., 2009. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. Sensors 9 (1), 568–601.

[31] Schmitt, R., Pavim, A., 2009. Flexible optical metrology strategies for the control and quality assurance of small series production. In: Optical Measurement Systems for Industrial Inspection VI. Vol. 7389. International Society for Optics and Photonics, p. 738902.

[32] Sciavicco, L., Siciliano, B., 2012. Modelling and control of robot manipulators. Springer Science & Business Media.

[33] Steen, S., 2011. Motstand og Propulsjon - Propell og Foilteori - TMR 4247 Kompendium. Department of Marine Technology, NTNU Trondheim.

[34] The Royal Institution of Naval Architects, 1978. Maritime Technology Monograph No. 6. ITTC Dictionary of Ship Hydrodynamics.

[35] Torr, P. H., Zisserman, A., 2000. Mlesac: A new robust estimator with application to estimating image geometry. Computer vision and image understanding 78 (1), 138–156.

[36] Varady, T., Martin, R. R., Cox, J., 1997. Reverse engineering of geometric models - an introduction. Computer-aided design 29 (4), 255–268.

[37] Wärtsilä Corporation, 2014. Fixed pitch propeller opti design. https://cdn.wartsila.com/docs/default-source/product-files/gears-propulsors/propellers/brochure-o-p-propeller-fpp-opti-design.pdf, accessed: 2018-05-22.

[38] Wu, D., Chen, T., Li, A., 2016. A high precision approach to calibrate a structured light vision sensor in a robot-based three-dimensional measurement system. Sensors 16 (9), 1388.

[39] Yeo, K. B., Choong, W. H., 2014. Marine propeller geometry characterization. Journal of Applied Sciences 14 (23), 3288–3293.

[40] Yin, S., Ren, Y., Guo, Y., Zhu, J., Yang, S., Ye, S., 2014. Development and calibration of an integrated 3d scanning system for high-accuracy large-scale metrology. Measurement 54, 65–76.

[41] Zhang, Z., 2000. A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence 22 (11), 1330–1334.

[42] Zivid Labs, 2018. Zivid 3D camera. `zividlabs.com`.

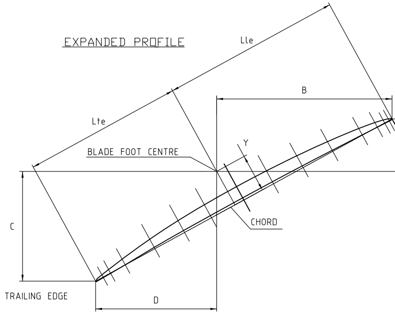
[43] Zivid Labs, 2018. Zivid data sheet. `https://www.zividlabs.com/uploads/documents/Zivid-3D-camera-Fact-sheet.pdf`, accessed: 2018-06-06.

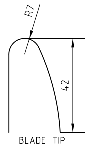
Appendix

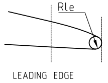
A.1 Propeller drawing

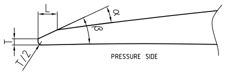
The propeller drawing is also included in the digital appendix. Sensitive information has been removed.

EXPANDED PROFILE



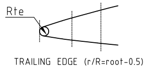






r/R	0.95	0.90	0.80	0.70	0.60	0.50
α	22.0	21.8	21.2	20.5	20.1	20.0
β	31.7	32.5	35.9	40.7	46.8	54.5
L	7.0	8.0	9.0	8.0	8.0	8.0
T	1.7	2.6	4.7	6.2	7.2	7.3

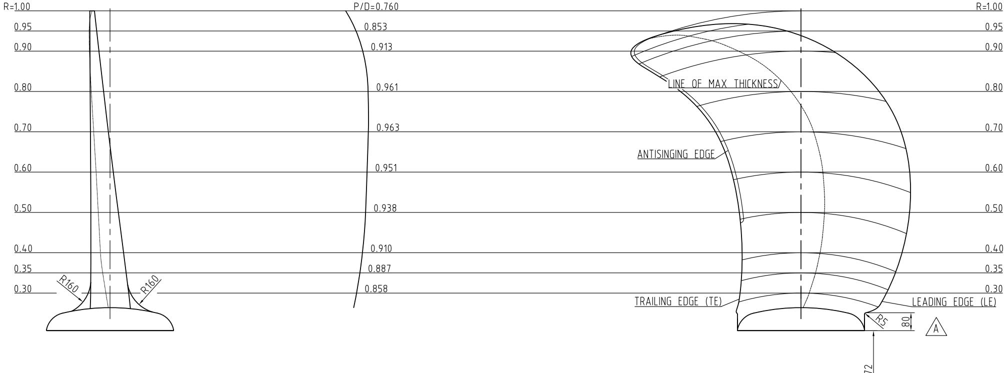
ANTI-SINGING EDGE (r/R=0.5-1.0)

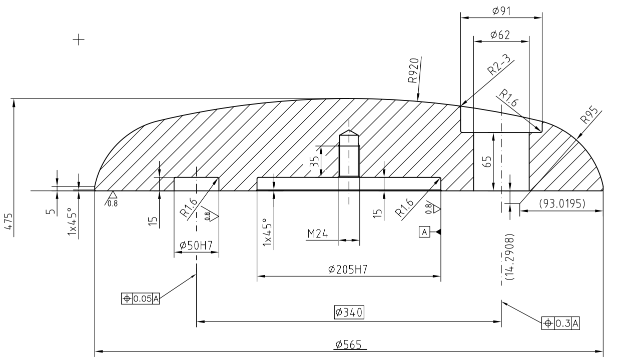


TRAILING EDGE (r/R=root-0.5)

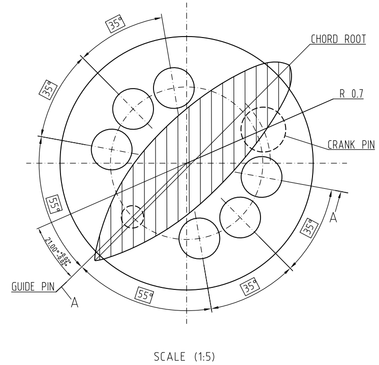
r/R	0.264	0.300	0.35	0.40	0.50	0.60	0.70	0.80	0.90	0.95	0.975	1.0
R	475.2	540.0	630.0	720.0	900.0	1080.0	1260.0	1440.0	1620.0	1710.0	1755.0	1800.0
P/D	0.832	0.858	0.887	0.910	0.938	0.951	0.963	0.961	0.913	0.853	0.811	0.760
P	2995	3089	3193	3276	3377	3424	3467	3460	3287	3071	2920	2736
α	45.09	42.31	38.89	35.91	30.84	26.77	23.65	20.93	17.90	15.95	14.83	13.60
Lle	316.5	354.5	398.6	434.5	483.5	500.4	476.4	383.1	155.1	-59.5	-234.8	-689.2
Lte	297.8	281.3	269.4	265.4	274.0	304.0	362.4	471.9	645.1	743.1	777.8	689.2
Ltot	614.3	635.8	668.0	699.9	757.5	804.4	838.8	855.0	800.1	683.6	543.0	0.0
Y	8.0	18.6	32.8	42.8	52.5	62.1	72.9	83.6	92.1	93.3	90.8	80.0
A	218.5	224.9	224.7	220.2	202.8	169.9	124.3	58.8	-40.0	-106.1	-147.9	-239.8
B	229.1	274.7	330.8	377.0	442.0	474.7	465.6	387.7	175.9	-31.6	-203.7	-651.1
C	216.6	203.1	194.7	190.3	185.5	192.4	212.2	246.7	285.9	293.9	286.9	239.8
D	204.6	195.5	189.1	189.9	208.3	243.4	302.7	410.9	585.6	688.9	728.7	651.1

r/R	c/C	TE	0.975	0.95	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.05	0.025	0.01	LE
ROOT 0.2639	SS	0.0	15.7	24.9	41.2	66.5	83.5	92.4	95.0	92.6	85.4	73.1	53.6	38.4	27.4	17.3	0.0
	PS	0.0	-13.9	-21.3	-34.3	-53.5	-67.1	-74.4	-76.6	-74.9	-69.5	-60.2	-45.3	-33.4	-24.5	-16.0	0.0
	T	R4.7	29.5	46.2	75.5	120.0	150.7	166.8	171.6	167.5	155.0	133.3	98.8	71.7	51.8	33.3	R10.3
0.30	SS	0.0	16.1	25.6	42.5	69.1	86.8	95.8	98.4	95.8	88.1	75.1	54.5	38.6	27.2	17.0	0.0
	PS	0.0	-12.8	-19.0	-29.8	-45.0	-56.2	-62.2	-64.0	-62.8	-58.5	-51.1	-39.1	-29.3	-21.8	-14.6	0.0
	T	R5.0	28.9	44.6	72.3	114.1	143.0	158.1	162.4	158.5	146.7	126.2	93.5	67.9	49.0	31.5	R9.7
0.35	SS	0.0	16.5	26.2	43.5	71.5	89.5	98.7	101.1	98.3	90.3	76.5	54.9	38.4	26.8	16.4	0.0
	PS	0.0	-11.4	-16.1	-24.2	-34.7	-43.1	-47.6	-48.9	-48.1	-45.2	-40.0	-31.5	-24.3	-18.5	-12.7	0.0
	T	R5.2	27.9	42.3	67.8	106.2	132.6	146.2	150.0	146.4	135.5	116.6	86.4	62.7	45.3	29.1	R9.0
0.40	SS	0.0	16.5	26.0	43.0	70.7	88.3	97.2	99.4	96.6	88.6	74.9	53.4	37.1	25.7	15.5	0.0
	PS	0.0	-10.6	-14.3	-20.5	-28.0	-34.3	-37.8	-38.7	-38.2	-36.1	-32.4	-26.2	-20.6	-16.1	-11.3	0.0
	T	R5.5	27.1	40.3	63.6	98.6	122.7	134.9	138.1	134.8	124.7	107.3	79.6	57.7	41.7	26.8	R8.3
0.50	SS	5.8	14.5	23.0	37.4	61.1	75.7	83.6	85.8	83.4	76.4	64.5	45.8	31.7	22.1	13.7	0.0
	PS	-5.8	-9.0	-12.1	-16.6	-21.5	-25.6	-28.5	-29.5	-29.2	-27.7	-25.1	-20.6	-16.5	-13.2	-9.7	0.0
	T	11.5	23.5	35.2	54.0	82.6	101.2	112.1	115.3	112.5	104.1	89.6	66.4	48.2	35.3	23.4	R5.8
0.60	SS	5.5	12.9	20.1	32.3	52.5	64.8	71.4	73.1	71.0	65.0	54.8	38.8	26.7	18.5	11.4	0.0
	PS	-5.5	-7.8	-10.0	-13.0	-15.8	-18.3	-20.3	-20.9	-20.8	-19.9	-18.3	-15.4	-12.6	-10.3	-7.7	0.0
	T	11.0	20.7	30.1	45.3	68.3	83.2	91.7	94.0	91.7	84.9	73.0	54.1	39.3	28.8	19.1	R4.7
0.70	SS	5.0	11.2	17.3	27.7	45.1	55.6	61.1	62.5	60.6	55.4	46.5	32.7	22.4	15.3	9.3	0.0
	PS	-5.0	-6.3	-7.5	-8.9	-9.3	-10.3	-11.2	-11.5	-11.6	-11.4	-10.9	-9.9	-8.5	-7.3	-5.7	0.0
	T	9.9	17.5	24.8	36.6	54.4	65.9	72.3	73.9	72.1	66.7	57.4	42.6	30.9	22.6	15.0	R3.7
0.80	SS	4.4	9.6	14.7	23.4	38.3	47.1	51.6	52.6	51.0	46.7	39.2	27.5	18.8	12.9	7.9	0.0
	PS	-4.4	-4.7	-5.1	-5.0	-3.2	-2.7	-2.7	-2.6	-3.1	-3.6	-4.3	-5.1	-5.2	-5.0	-4.4	0.0
	T	8.7	14.3	19.7	28.4	41.5	49.8	54.3	55.2	54.1	50.3	43.5	32.6	24.0	17.8	12.2	R3.0
0.90	SS	3.1	7.2	11.2	18.2	30.2	37.2	40.8	41.5	40.3	36.9	30.9	21.5	14.6	10.0	6.1	0.0
	PS	-3.1	-2.8	-2.5	-1.4	1.8	3.2	3.7	3.9	3.4	2.4	0.9	-1.2	-2.3	-2.8	-2.9	0.0
	T	6.3	10.1	13.7	19.6	28.4	34.0	37.1	37.6	37.0	34.5	30.0	22.7	16.9	12.7	8.9	R2.2
0.950	SS	2.5	5.7	8.9	14.5	24.1	29.8	32.6	33.2	32.3	29.5	24.8	17.3	11.8	8.1	5.0	0.0
	PS	-2.5	-2.1	-1.8	-0.8	2.0	3.3	3.8	4.0	3.5	2.6	1.2	-0.7	-1.7	-2.2	-2.4	0.0
	T	4.9	7.9	10.7	15.3	22.1	26.4	28.8	29.2	28.8	27.0	23.6	18.0	13.6	10.4	7.4	R1.8
0.975	SS	2.2	4.7	7.2	11.6	19.0	23.3	25.5	26.0	25.3	23.2	19.6	13.9	9.6	6.7	4.3	0.0
	PS	-2.2	-2.1	-2.1	-1.6	-0.1	0.5	0.7	0.9	0.5	-0.1	-0.9	-1.9	-2.3	-2.5	-2.4	0.0
	T	4.3	6.8	9.3	13.2	19.1	22.8	24.8	25.1	24.8	23.3	20.5	15.7	11.9	9.2	6.7	R1.7
1.000									21.0								





SECTION A-A (1:2.5)



SCALE (1:5)

STATIC BALANCING

THE BLADES IS TO BE BALANCED TO ISO484/2 - CLASS 1
PERMITTED DIFFERENCE OF TORQUE BETWEEN THE BLADES IS 75 Nm
TORQUE = Gblade × tpr (tpr = distance prop center - center of gravity)
MAX BALANCE WEIGHT AT TIP 4.25 kg

PROPELLER DATA

DIAMETER OF PROPELLER = 3600 mm
PITCH RATIO AT 0.7 R = 0.963
MEAN PITCH RATIO -Pm/D = 0.918
BLADE AREA RATIO -EAR = 0.389
NUMBER OF BLADES -Z = 4
HUB DIAMETER = 950 mm
DIRECTION OF ROTATION
SEEN FROM AFT = RIGHTHANDED
MOMENT OF INERTIA BLADES
AND HUB IN AIR = 2600 kgm2

SKEW ANGLE = 29°

CALCULATED WEIGHT READY MACHINED = 649 kg

THE BLADE TO BE MANUFACTURED
ACC. TO : ISO484 Class I

CLASSIFICATION SOCIETY:
GL - ice E3

Surface texture ISO/R 1302 Ra µm √ ∇ ∇

When not otherwise stated :
Angle dimension tolerance shall comply with SS-ISO 2768-1 middle column.
Linear dimension tolerance shall comply with SS-ISO 2768-1 middle column. (See the table below)

Basic size	Tolerances	Basic size	Tolerances
(0,5)- 3	±0,1	(1000)- 2000	±1,2
(3)- 6	±0,1	(2000)- 4000	±2
(6)- 30	±0,2	(4000)- 8000	±3
(30)- 120	±0,3	(8000)- 12000	±4
(120)- 400	±0,5	(12000)- 16000	±5
(400)- 1000	±0,8	(16000)- 20000	±6

A.2 Article accepted for SYROCO 2018 conference

As part of this project, an article was accepted for SYROCO 2018 conference. This article is included in the digital appendix.

A.3 Code

The program taking a point cloud of a propeller blade as input and returning the design parameters in a csv file is shown below. Other code can be found in the digital appendix, as well as .ply files from the scans.

```
function [datatable] = propellerparameters(pointcloud, design_diameter,roR)
% Function taking a point cloud of a controllabel pitch propeller blade
% as an input, and returning the propeller design parameters for the
% sections specified.
%
% Inputs:
% pointcloud - point cloud of a propeller blade in .ply format. The point
% cloud needs to be oriented correctly to the coordinate system. Ex.
% 'zividscan.ply'
%
% design_diameter - the design diameter of the propeller in mm, either
% given in a prop drawing or measured
%
% roR - the radial sections where design parameters are wanted, given as a
% fraction of propeller radius. Default: [0.4,0.5,0.6,0.7,0.8,0.9,0.95]
%
% Output:
% datatable - A table with the design parameters for the propeller
% a csv-file with the design parameters are also created

    format shortG

    if nargin == 2
        roR = [0.4,0.5,0.6,0.7,0.8,0.9,0.95];
    end

    ptcloud = pcread(pointcloud);
    X = ptcloud.Location(:,1);
    Y = ptcloud.Location(:,2);
    Z = ptcloud.Location(:,3);

    points = [X,Y,Z];
    numberofpts = size(points,1);

    % plot propeller point cloud
    ptcloudfig = figure;
    pcshow(points)
    hold on

    numberofparams = 13;
    datatable = zeros(numberofparams,size(roR,2));

    pointdist = zeros(numberofpts,1);
    for i = 1:numberofpts
        pointdist(i) = sqrt(Y(i)^2 + Z(i)^2);
    end

    [radius,indice] = max(pointdist);
    threshold = radius/10000;

    % for the radii specified
    for j = 1:size(roR,2)

        section = zeros(numberofpts,3);
        sectionradius = roR(j);

        % for all points in pointcloud
        for i = 1:numberofpts

            % if points are within the threshold distance from a radial
            % cylinder
            if  (sqrt(Y(i)^2 + Z(i)^2) <= sectionradius*radius+threshold) ...
                    && (sqrt(Y(i)^2 + Z(i)^2) >= sectionradius*radius -...
                    threshold) && (Z(i) >= 450)
                % the points that are close to the cylinder fit is saved
                section(i,:) = [X(i),Y(i),Z(i)];
            else
                section(i,:) = 0;
            end
        end

        % make a cleaner cloud of the section by removing points in the origin
        [row,col] = find(section);
        sectionpts = size(row,1);
        sectioncl = zeros(sectionpts,3);

        for i = 1:sectionpts
                sectioncl(i,:) = section(row(i),:);
        end

        % plot section
        figure(ptcloudfig);
        hold on
        scatter3(sectioncl(:,1),sectioncl(:,2),sectioncl(:,3),'.','k');

        %% opening out section
        % variables
        spline = sectioncl;
        splinepts = size(spline);
        r1 = zeros(splinepts(1),1);
        x2 = spline(:,1); % (= x1)
        y2 = zeros(splinepts(1),1);
        z2 = r1;

        x1 = spline(:,1);
        y1 = spline(:,2);
        z1 = spline(:,3);
        phi1 = zeros(splinepts(1),1);

        % for all points in section
        for i = 1:splinepts(1)
            r1(i) = sqrt(y1(i)^2+z1(i)^2);
            z2(i) = r1(i);

            if y1(i) >= 0
                phi1(i) = asind(z1(i)/r1(i));
                y2(i) = ((90-phi1(i))/360)*r1(i)*2*pi;
            elseif y1(i) < 0
                phi1(i) = acosd(z1(i)/r1(i));
                y2(i) = -(phi1(i)/360)*r1(i)*2*pi;
            end
        end

        uwspline = [x2,y2,z2];
        %% find the edges of the section

        sectionsumy = sum(uwspline(:,2));
        sectionsumx = sum(uwspline(:,1));

        centroid = [-sectionsumy/sectionpts,sectionsumx/sectionpts];

        trailingpts = zeros(sectionpts,1);
        leadingpts = zeros(sectionpts,1);

        for i = 1:sectionpts

            if -y2(i) >= centroid(1)
                leadingpts(i) = sqrt((-y2(i)-centroid(1))^2+(x2(i) -...
                    centroid(2))^2);
            elseif -y2(i) <= centroid(1)
                trailingpts(i) = sqrt((-y2(i)-centroid(1))^2+(x2(i) -...
                    centroid(2))^2);
            end

        end

        [leadinglength,leadingindex] = max(leadingpts);
        [trailinglength,trailingindex] = max(trailingpts);
        hold on
        edges = [-y2(leadingindex),x2(leadingindex);
            -y2(trailingindex),x2(trailingindex)];

        %% calculate design parameters
        L = sqrt((edges(1,1)-edges(2,1))^2+(edges(1,2)-edges(2,2))^2);
        pitchsection = atand((edges(1,2)-edges(2,2))/(edges(1,1) -...
            edges(2,1)));
        Pitchlength = tand(pitchsection)*(pi*roR(j)*design_diameter);

        A = edges(1,2);
        B = edges(1,1);
        C = -edges(2,2);
        D = -edges(2,1);
```

```
            Leadingangle = atan2(A,B) + (pi/2-deg2rad(
                pitchsection));
            Lle = sin(Leadingangle)*(sqrt(A^2+B^2));
            Lte = L - Lle;
            Yval = Lle/tan(Leadingangle);

            % make table with design parameters
            datatable(1,j) = sectionradius;
            datatable(2,j) = sectionradius*design_diameter/2;
            datatable(3,j) = Pitchlength/design_diameter;
            datatable(4,j) = Pitchlength;
            datatable(5,j) = pitchsection;
            datatable(6,j) = Lle;
            datatable(7,j) = Lte;
            datatable(8,j) = L;
            datatable(9,j) = Yval;
            datatable(10,j) = A;
            datatable(11,j) = B;
            datatable(12,j) = C;
            datatable(13,j) = D;
        end

    % write csv file to open in excel
    csvfile = strtok(pointcloud,'.');
    csvwrite(strcat(csvfile,'.csv'),datatable);
end
```