Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Håvard Skauen

Machine learning based digital twin framework for aquaculture net cage system

Master's thesis in Civil and Environmental Engineering
Supervisor: Guomin Ji
Co-supervisor: Biao Su & Xu Cheng
June 2022

NTNU
Norwegian University of Science and Technology

Håvard Skauen

Machine learning based digital twin framework for aquaculture net cage system

Master's thesis in Civil and Environmental Engineering
Supervisor: Guomin Ji
Co-supervisor: Biao Su & Xu Cheng
June 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

In this master's thesis, a framework for digital twin for fish farming facilities using numerical simulation, machine learning and sensor data is presented. The numerical simulations are performed with FhSim which is a software created by SINTEF Ocean. The net cage system that is simulated in FhSim is the CAC project which is a research project owned by MOWI. The machine learning algorithm used is XGBoost. With FhSim, 640 simulations were performed in this master's thesis. These simulations were run with different current velocities for different depths in the same direction. By training the machine learning model with the result data generated by FhSim, predictions of the mooring connection forces in the net cage system bridles could be made. The accuracy of the machine learning model has varied, but with a training/testing data split of 80/20 in XGBoost, the average uncertainty was 0.5 %. The thesis further suggests how this framework for the digital twin can be further developed into an accurate digital twin model. This presupposes that the machine learning model can be trained with more generated data from FhSim. The digital twin can then be used to map the wear of the structural components in the net cage system so that accidents or unnecessary replacement of components is avoided.

Sammendrag

I denne master oppgaven er det presentert et rammeverk for digital tvilling for fiskeoppdrettsanlegg ved å bruke numerisk simulering, maskin læring og sensor data. De numeriske simuleringene er utført med programvaren FhSim som er utarbeidet ved SINTEF Ocean. Fiskeoppdrettsanlegget som er simulert i FhSim er CAC prosjektet til MOWI. Maskin lærings algoritmen som er brukt er XGBoost. Med FhSim ble det gjennomført 640 simuleringer i denne master oppgaven. Ved å trene maskinlæringsmodellen med denne resultatdataen fra FhSim har det vært mulig å spå fortøyningskreftene i tøylene i fiskeoppdrettsanlegget. Nøyaktighetsgraden på maskinlæringsmodellen har variert, men med en trening/testing data split på 80/20 i XGBoost var den gjennomsnittlige usikkerheten på 0.5%. Oppgaven foreslår videre hvordan dette rammeverket for digital tvilling kan videreutvikles til å utarbeide en nøyaktig digital tvilling modell. Dette forutsetter da at maskinlæringsmodellen kan trenes med mer data generert med FhSim. Da kan denne digitale tvillingen brukes til å kartlegge slitasje av strukturelle komponenter i fiskeoppdrettsanlegget slik at ulykker eller unødvendig utskifting av komponenter unngås.

Preface

This thesis concludes my Master of Science degree in Civil and Environmental engineering at the Norwegian University of Science and Technology (NTNU). Although the topic of sea loads and machine learning are topics beyond most of what I have worked with earlier in the master's degree. The research performed in this master's thesis has proved to be very instructive. It has been very interesting to see an idea come to life and eventually being tested and proven. What I find very exciting about this master's thesis is what kind of future research questions will arise from the findings made in this thesis.

During my thesis work, I have been supervised by Guomin Ji, Ph.D. (NTNU), Xu Cheng, Ph.D. (IEEE) and Biao Su, Ph.D. (SINTEF Ocean) and I would like to thank them all for assisting me during this master's thesis.

I would like to express my gratitude for Guomin for his help in arranging this master's thesis and guidance during my degree. Guomin has guided me both through the master's thesis and he was also my supervisor during my bachelor thesis. He also gave me the opportunity to be a student assistant in one of his courses. Guomin has guided me through both the bachelor's and master's degrees and I am very grateful for all the help and guidance he has given me over the last few years.

I would also like to thank Cheng for helping me with the machine learning section of the thesis. Had it not been for Cheng, the machine learning part would have been a lot more difficult and time consuming.

Further I want to thank Biao for giving me the opportunity to write the master's thesis in collaboration with SINTEF Ocean. Biao has helped to develop the research question for this thesis and has also assisted with the guidance of FhSim. As well as helping with other questions that have arisen along the way.

Lastly I would like to thank my family for their support during my master's degree. My my father Rein O. Skauen, my mother Hilde M. Skauen, and my sister Cathrine Skauen.

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Background and motivation

The human population is increasing and has now reached 7.8 billion in 2021 and will continue to grow for years to come. With the current growth of the world population, global food demand will also increase. From 1961-2017 the average annual growth rate of fish consumption outpaced annual population growth rate with the annual fish consumption being 3.1% and the annual population growth being 1.6% [1]. The average annual fish consumption also outpaced the average annual consumption of all other animal proteins, with the exception of poultry [1].

Though it's not only the food demand that makes fisheries and aquaculture important. In the world, the livelihoods of 60 million people are already dependent on fishing and aquaculture [2]. In the period from 2001-2018 the world aquaculture production of farmed aquatic animals has had an average growth of 5.3% per year [1].

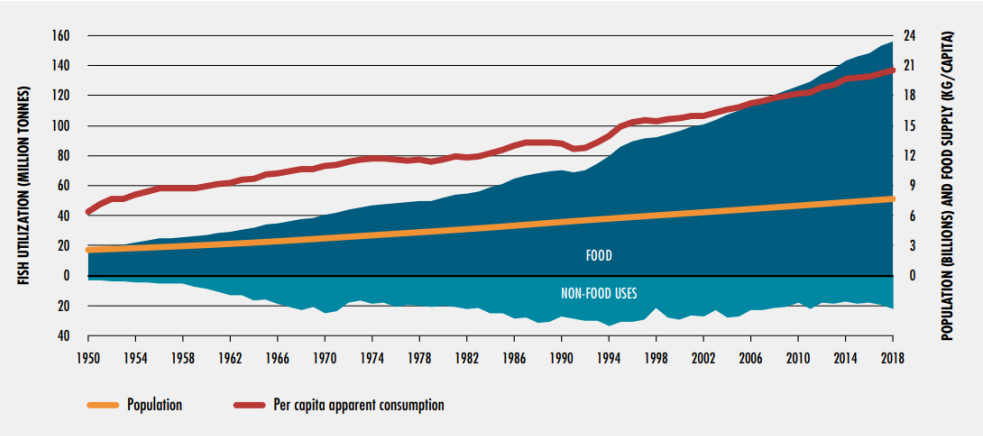


Figure 1.1: FOA Fish Consumption [1].

In 2010, the Food and Agriculture Organization of the United Nations reported that 44 million people were directly engaged in the capture of fisheries and that this was an increase from 1980 where 16.7 million people was engaged [3]. This means that the number of people that have their livelihoods dependent on fishing and aquaculture has increased from 16.7 million to 60 million over the last 38 years. With the increasing level of fisheries and aquaculture, the space for near shore aquaculture facilities will also decrease. This leads to a demand on these facilities to be as efficient and durable as possible.

According to Norwegian law "the NYTEK regulations", the owner of an aquaculture facility is responsible for ensuring that the facility is in a sound technical condition at all times [4]. Currently the maintenance of fish farms is done based on experience and inspections. NS 9415, 15.7 describes that all inspections and maintenance must be summarized in detailed inspection and maintenance plans [5]. The Norwegian fish farming industry thus has a system for maintaining fish farming, but lacks technology to be able to accurately predict the condition of the fish farming facility at all times.

This thesis looks into the possibility of using existing technology to be able to predict maintenance of components in net cage systems by introducing a framework for a digital twin. AKVA Group has made an electrical sensor buoy. This buoy has sensors that measure the oceans current direction and velocity, wave height and direction, oxygen, temperature and salinity levels. This buoy can lead to predictions based on experience becoming more accurate. Although experience-based predictions can work moderately well, knowing the exact force distribution in the net cage system will be the most optimal solution for estimating maintenance of the components. A good solution is therefore to carry out numerical simulations, with the environmental loads registered by the sensor buoy made by AKVA Group. This can be done with FhSim. This is a software developed by SINTEF OCEAN that makes it possible to perform numerical simulations on different kinds of net cage systems. The problem however is that the numerical simulations take longer than real time to complete. This means that it's not possible to know the force distribution at the fish farm in real time with numerical simulations.

To solve this problem a solution combining numerical simulation and machine learning can be used. With an algorithm that predicts the force distribution in the net cage system without the necessary simulation time, maintenance of the components in the net cage system can be mapped into a maintenance pattern. In this thesis, the net cage system with which numerical simulations have been performed, is the facility on the CAC Project. This is because by during collaboration with AKVA Group during this thesis has provided access to an active sensor buoy located outside the the CAC net cage system.

1.2 Scope and limitations

The aim of this thesis is to show that stored data from numerical simulations can be combined with machine learning to quickly generate real-time data of the environmental ocean force distribution on a net cage system. This can lead to further development for a digital twin framework. The numerical simulations were performed on the CAC net cage system. This fish farm, has an angled net cage type called spaghetti cage. In this thesis, numerical simulations with FhSim have been carried out for this type of net cage. Only one of the cages on the CAC project has been included in the numerical simulations. It is also important to state that there are no fish in the net for the simulations in this thesis. The purpose of these reductions is to simplify the simulation model and to show that it is possible to develop a solution with machine learning rather than running an unnecessarily large number of detailed simulations.

In order to show that a machine learning solution can be realistic, large amounts of data is needed for training the machine learning model. Simplifications have therefore been made to the parameters of the input data of the numerical simulations. These simplifications consists of reducing the environmental forces that affects the net cage system. This is done so that the quantity of simulations increase, though the time spent simulating remains the same. The reduction consists of removing waves from the numerical simulations. The current is also simulated in the same direction, but with different velocities at different depths. When choosing the current direction for the numerical simulation, symmetry was taken into account. The numerical simulations were completed for directions 1.57 radians to 3.14 radians. With symmetry these directions can be used for the remaining directions of the net cage system.

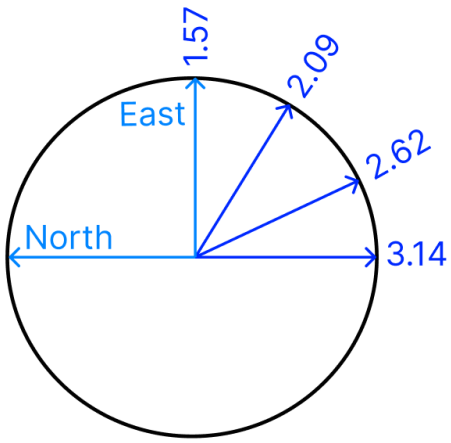


Figure 1.2: Current directions.

Limits have also been established in the numerical simulations for current velocities. The choice of current velocities originates from AKVA Group's sensor buoy's registered data. Current velocities between 0.0 m/s and 0.9 m/s were used in the simulations for the upper part of current. The lower part of the current were simulated in a range from 0.0 m/s to 1.5 m/s. In the result section for machine learning, the thesis focuses on the environmental force distribution on the bridles in the net cage system.

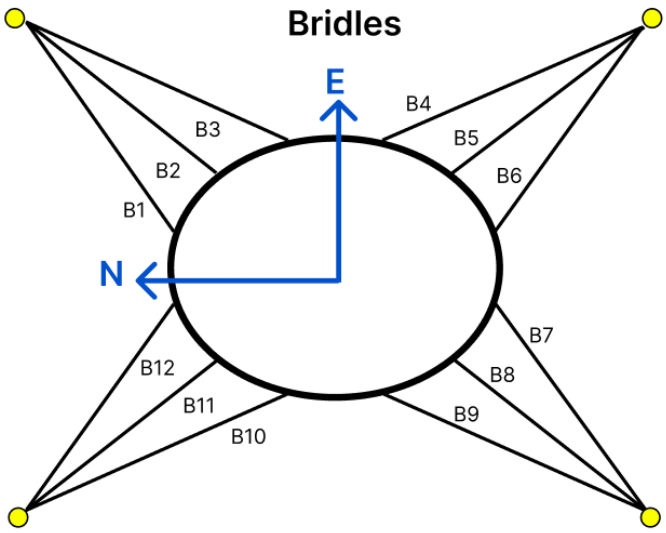


Figure 1.3: Bridles.

1.3 Structure of the thesis

Chapter 1, Introduction provides an introduction to how important the fish farming industry in the world has become. This chapter also includes how the increasing world population requires optimization of the worlds aquaculture and how this can be done using existing technology. The chapter also covers scope and limitations made for this thesis.

Chapter 2, Methodology provides the methods used in this thesis.

Chapter 3, Theory provides the necessary theory for this thesis. This includes the general structure of net cage systems, as well as important hydrodynamic properties and models. The chapter also includes the theory behind FhSim and how the numerical simulation program is set up. Furthermore, the chapter covers general theory of machine learning as well as the theory behind the chosen machine learning algorithm used in this thesis. Finally, the theory behind AVKA Group's sensor buoy is included.

Chapter 4, CAC case study shows the selected case study used in this thesis. The chapter also explains exactly why this case was chosen. Furthermore, the chapter covers the specific properties of the net cage system on the CAC project and shows these, as well as the design of the net cage.

Chapter 5, Numerical simulation covers the numerical simulation performed in this thesis. This includes the configuration of FhSim and the configuration for the CAC projects net cage system in FhSim. Further the chapter previews how the simulations were performed and how the output data was stored and managed. Finally, the chapter shows the end result data of the simulations.

Chapter 6, XGBoost algorithm previews the XGBoost algorithm setup for this thesis. This includes different parameters of the code and how they affect the machine learning model. The chapter also shows the results the machine learning model has generated.

Chapter 7, AKVA Group sensor data previews a small sample of the AKVA Groups buoy sensor data registered from 01.08.2021 to 31.08.2021. This data is previewed with graphs for the current direction and the current velocity for this month.

Chapter 8, Digital twin proposal shows how the machine learning model can be combined with the registered data from the sensor buoy, to predict mooring connection force in the CAC net cage system caused by the current and it's velocity and direction. The chapter includes the prediction and true forces of a case registered from the buoy sensor.

Chapter 9, Discussion includes discussions on the FhSim simulation result and the machine learning results. The chapter also discusses the exclusion of waves in the FhSim model and how the quantity of simulations could have been reduced. Further the chapter includes discussion of reducing faulty data registered by the buoy sensor. Lastly the chapter covers discussion on the use of machine learning for aquaculture net cage systems and the quality of the simulation output generated with FhSim in this thesis.

Chapter 10, Conclusion concludes the work done in this thesis. At the same time, the chapter justifies whether such technology is useful for the aquaculture industry in Norway and the world.

Chapter 11, Further work covers the further work which can be done based on the findings in this thesis. This includes looking into advanced current models in FhSim for net cage systems. The chapter also covers further work for simulations that can be started with an accurate starting position of where the last simulation ended. The chapter suggests a method of reversing the machine learning model as well. Further the chapter comments on the optimization of FhSim for use in machine learning models. Lastly the chapter ends with a proposition for future simulations that will produce better machine learning models to be combined with the AKVA Group sensor buoy.

Appendix A, Simulation result includes hyperlinks to stored data sheets for both raw FhSim output data and resultant calculated FhSim output data.

Appendix B, Python scrips includes the python scripts used to manage the raw data generated with FhSim and also the script used to calculate the resultant forces of the bridles. The appendix also previews the script used for the XGBoost algorithm.

Appendix C, XGBoost results includes a hyperlink to the stored XGBoost result data. This data includes predictions for each bridle using 20% of the total data in each model.

Chapter 2

Methodology

This chapter contains the methods used to investigate whether a digital twin framework can be developed via numerical simulation, sensor data and machine learning in this thesis.

2.1 Literature study

To get a better understanding about aquaculture net cage systems, numerical simulation, machine learning and the aquaculture industry in Norway, a literature study has been conducted for this thesis. This literature study has consisted of reviewing books on aquaculture and hydrodynamics, scientific research papers and previously written assignments of net cage systems.

2.2 Collaboration with the industry

This master thesis has been written in collaboration with companies in the aquaculture industry in Norway. The collaborating companies for this master's thesis other than SINTEF Ocean is AKVA Group and MOWI. AKVA Group is a company who makes solutions and performs services for the aquaculture industry. AKVA Group provided an insight into the technological shortcomings for typical net cage systems. MOWI is one of the world's largest companies within fish farming and has provided the consumer's perspective of what is important for an aquaculture facility.

2.3 Case study CAC

In this thesis, a case study has been carried out for an aquaculture facillity at Vindsvik in Hjelmeland. This is because AKVA Group has experimented with a working sensor buoy on a project at the Center for Aquaculture Competence. This is a project owned by MOWI. This led to the net cage system at the CAC project becoming very relevant. This was because the CAC project could make it possible to combine sensor buoy technology with a net cage system. MOWI has provided full access to both sensor buoy data, as well as detailed data about the design of the net cage at the project.

2.4 Software for numerical simulation

In this thesis, FhSim is the software that has been used for the numerical simulation. SINTEF Ocean has developed FhSim and is also the institution this master's thesis is written in collaboration with.. In this thesis 640 simulations in FhSim have been completed, where each simulation had a completion time of around five hours. This resulted in 160 GB of output data.

2.5 Implementation of numerical simulation

To be able to perform the required number of simulations for machine learning, external virtual machines was used. This was done by using Microsoft Azure. In Azure, virtual machines from Japan, Australia, Switzerland and India was used. This made it possible to perform a larger number of numerical simulations simultaneously. The completed simulations were stored online via a cloud solution storage system. The result files were named with the input values of the simulation.

2.6 Data processing

The output data files were approximately 0.25 GB per file. This output data stored as a csv-file. These kinds csv-files with this amount of size took way to long to open manually. In each file, lots of output data was generated about the entire net cage system, but not everything was meant to be included in the machine learning model. Instead of changing the code of FhSim to reduce the inclusions of the data. A script created in python was therefore used to automate the calculation of resultant forces for Bridles 1 to 12.

2.7 Sensor buoy data

In addition to having made the technological sensor buoy, AKVA Group has created a cloud based storing system for data stored by the buoy. This is how the result data from the buoy was collected for this thesis. The sensor data that has been the foundation for the simulations in this thesis was downloaded in the range from July 2021 to February 2022.

2.8 Machine learning algorithm

For the machine learning part of this thesis. The machine learning algorithm used was XGBoost. XGBoost was chosen as the algorithm because of it's reputation for accuracy and efficiency. It is also known for being easy to set up. This was beneficial because this task addresses many topics, and a very advanced and complex algorithm would not have been appropriate with regard to the time frame.

Chapter 3

Theory

This chapter provides the necessary theory to understand the structure of traditional net cage systems. That includes various cage collars, weights, net bags, mooring systems, fixing points and buoy types. Different structure and material choices is also elaborated.

Further the chapter provides information around important hydrodynamic properties like the Reynolds number and the solidity. The chapter also includes hydrodynamic models as well. In this section the Morison model and screen force model is included. The chapter also includes theory behind FhSim and how the simulation software is set up. That means explainations on how the calculations for the cage collar, net, cables and bottom weight are made.

Theory behind machine learning is also included in this chapter. This includes different machine learning types and a more detailed explanation of the theory behind the machine learning model used in this thesis. In the section on the machine learning model used in this thesis the theory behind the XGBoost algorithm is included. The specifications of the AKVA Group sensor buoy is also included in the theory chapter. This section consists of the parts the buoy is put together by, and how the buoy's mooring system is set up.

3.1 Net cage structure

In Figure 3.1, the major components in a traditional net cage system is displayed. This consists of a net bag/cage, a jumping net, a cage collar and a mooring system. The net bag/cage has a weight at the bottom which leads to the net bag/cage being able to maintain an acceptable volume. The jumping net at the top is there so that the salmon is not able to jump out of the net bag/cage. The cage collar is there to spread out the net bag/cage, while at the same time leading to increased buoyancy, keeping the net bag/cage in its intended position. The last major component is the mooring system. This ensures that the facility stays in its intended position [6].

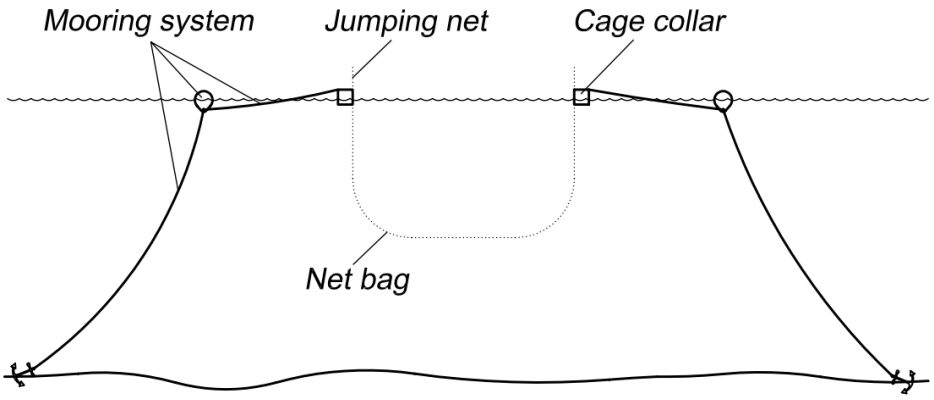


Figure 3.1: Aquaculture net cage structure [6].

3.1.1 Cage collar

The cage collar is very important for a net cage system. This is because it keeps the net cage system afloat and keeps the cage in its intended position. The cage collar also leads to the net being able to retain the right shape, and it can also be used as a work platform. Expanded polystyrene (PS) is usually used for buoyancy which is placed in a cylindrical ring of polyethylene (PE) for protection from sunlight. If polyethylene (PE) is not used, the polyesters (PS) will age from the sunlight and eventually become brittle. Another danger of using the polystyrene (PS) without protection is that fouling can occur [6]. Fouling describes marine organisms that grow on partially submerged marine structures [7].

At the same time it's important that there's not too much buoyancy. This will lead to a more expensive design, and will also cause the cage collar to float very high in the water. When the cage collar floats high in the water it will follow the wave movements more. Following the wave movements will lead to increased induced vertical motion which again will lead to increased strain for both the net and the mooring system. This will have increased consequences if wave activity increases in the area.

The reason why circular collars are used is because they have the best distribution of forces. The forces that affect the collar are equal around the entire circumference. Cylindrical polyethylene molds are also used because of the improved aerodynamic properties, this also results in a reduction in the transmission of the current force. At the same time, it is important to note that the force caused by the current has very small influence on the collar compared to how it affects the net [6].

Figure 3.2: Cage collar [8].

3.1.2 Weights

Current causes the volume in the cage to be reduced. To combat this a method of using weights at the bottom of the net cage is used to maintain the net volume. Although this is a good solution to keep as much volume as possible, it's important not to add too much weight. This is because more weight leads to greater current force on the net bag and its connections to the single cage system [6].

It's also common to use a ring as a weight for large individual cages. This ring is called a sinkertube. The sinkertube is made of the same fabric as the outer layer of cage collar polyethylene (PE), that is high-density polyethylene (HDPE). This sinkertube made of of HDPE is then filled with weights and then attached to the bottom of the net cage to reduce the volume change of the net cage [6].

Figure 3.3: Circular pen [8].

Figure 3.4: Angled pen [8].

3.1.3 Net bags

The most common material to use for net cages is synthetic plastic materials. This can be polyamide (PA, nylon) predominate. Polyethylene (PE) used for cage collars is also used for net cages because it has greater resistance to fouling, but it's stiffer to work with. Nylon is more common because it's cheaper, strong and not as stiff as polyethylene (PE) [6]. Nets can be knotted or they can be sewn together. The normal mesh shape is square.

To understand how the net behaves in the ocean and how much the net is stretched, an expression for the hanging ratio can be used. This ratio is given by the formula: $E = L_x/L_y$. Where L_x is the length of the line where the net is fixed and L_y is the length of stretched net panel. The E value of net cages used for net cage systems are normally between 0.6-0.9 [6].

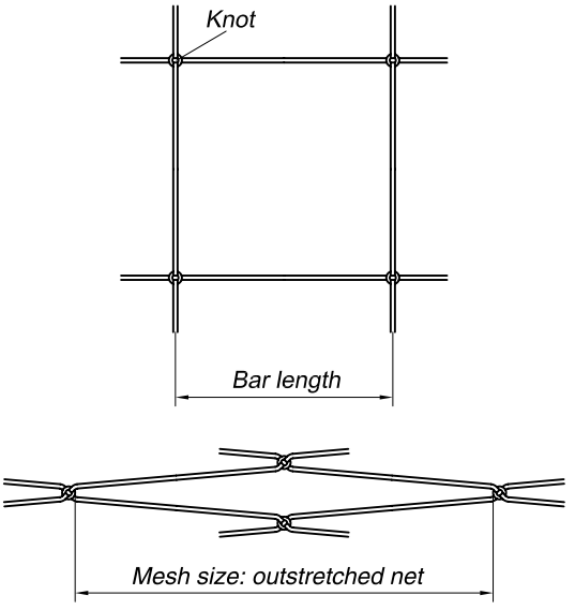


Figure 3.5: Mesh dimensions [6].

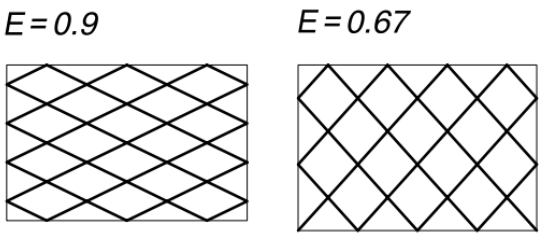


Figure 3.6: Hangingratio[6].

Another important parameter is the **solidity ratio**. This unit describes the ratio between the area covered by the lines and the total area of the net. This value is important when calculating the flow of water through the net. Solidity ratio is given by the sign S_n and is usually between 0.1-0.3 for clean nets. If the nets have been exposed to fouling, the solidity ratio could be higher [6]. More about solidity in chapter 3.2. In Norway the normal lifetime of a net cage is 5 years and the recommended depths for the nets are 0.8-1.25 times the net bags diameter [6].

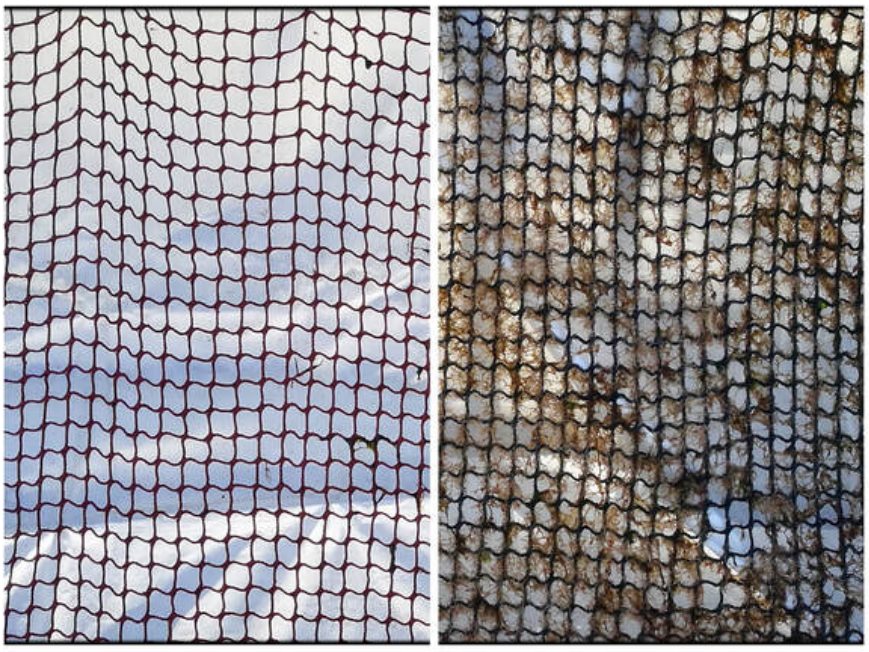


Figure 3.7: Net fouling [9].

3.1.4 Mooring system

The mooring system is important because it keeps the net cage system in its specific position while limiting the transmission of forces, especially the vertical forces to the net bag/cage. There are two common design systems used for mooring. This is pre-stressed and slack mooring [6].

Pre-stressing is done when there is high tide. With net cage systems being a flexible construction, the best mooring system is the pre-stressed design. This is because the pre-stressed mooring system will lead to the forces on the cage collar getting distributed evenly [6]. This pre-stressed mooring system consists of three main components. This is the mooring lines, buoys and the anchors. The mooring lines also have point of attachment to the cages [6]. A standard single net cage is usually moored with either 4 or 6 buoys connected by mooring lines to the anchor and to the net cage.

It's usually recommended that the mooring line attached to the anchor is divided into either two or three new lines before it's connected to the cage collar. The reason why this is a recommended solution is because it's in these contact points that the forces are transmitted, and therefore also important to distribute the forces in these points of contact [6].

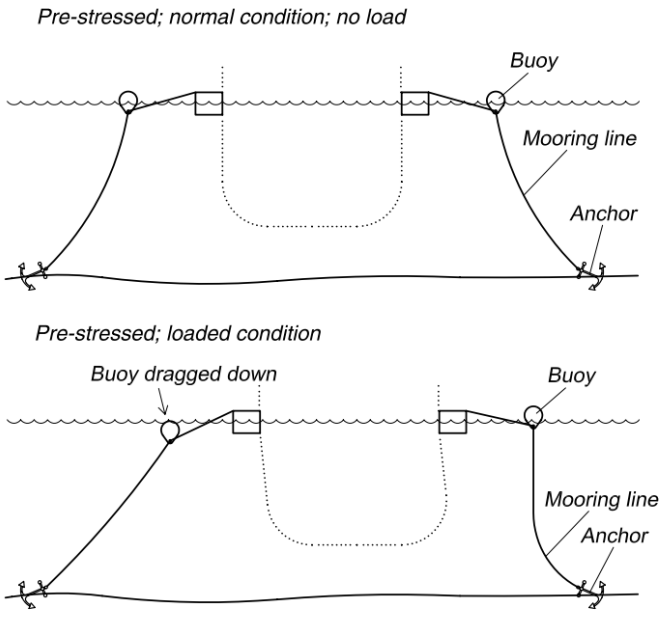


Figure 3.8: Pre stressed mooring [6].

3.1.5 Fixing point

Like mentioned in 3.1.4, the mooring lines are split into more lines before connecting to the cage collar. This connection point on the cage collar consists of a shackle connected to the line split from the mooring line. The force that is important to distribute with several connection points is the vertical force.

The vertical force is a component of the force that is transmitted to the connection point. It's also a horizontal force component, but its not as exposed to the tide. This way of distributing the forces with several connection points to the cage collar is called hen foot mooring [6].

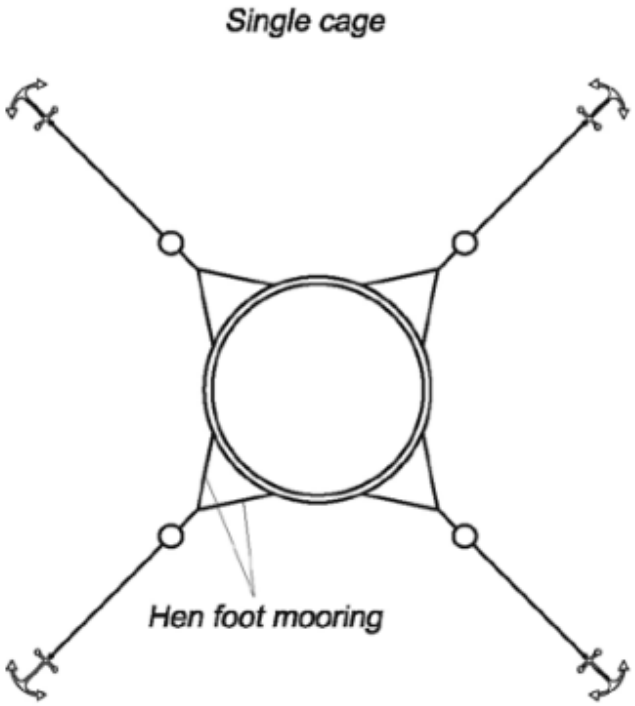


Figure 3.9: Single cage mooring [6].

3.1.6 Mooring lines

Synthetic ropes are widely used for mooring lines in the fish farming industry. Synthetic ropes can be made of different materials. Some materials used for net cage systems are Polyamide (PA, Nylon), Polyethylene (PE), Polyester (PES) and Polypropylene (PP). The reason why different types of material are used is because of their properties. The service life of such synthetic ropes can vary depending on how much force it is exposed to and what kind of material is used, but the starting point is four years [6].

It is also possible to use materials such as chains. Chains can withstand stronger forces better than a synthetic rope, but at the same time it's prone to corrosion, has more weight and has a higher cost. Therefore, the chain is usually used more often for the first 15-20 meters of the mooring lines [6].

3.1.7 Buoys

Buoys are used to reduce the vertical forces on the cage collar as much as possible and are also used to pre-stress the single cage. The most common buoys used for open sea cages are between 200 liters to 700 liters, and is shaped round and cylindrical. The buoys are usually filled with expanded Polystyrene(EPS) or Polyurethane(PU) [6].

Figure 3.10: Buoy [10].

3.1.8 Anchors

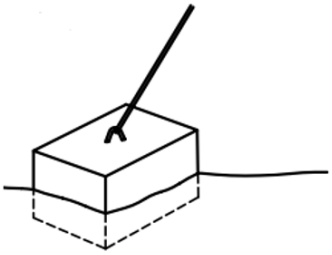


Figure 3.11: Block anchor [6].

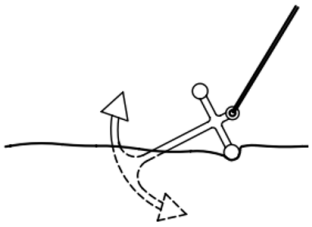


Figure 3.12: Drag anchor [6].

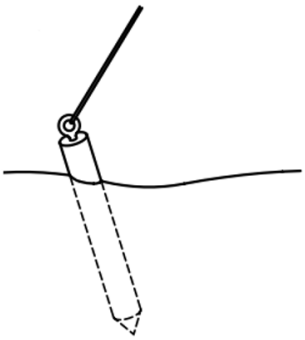


Figure 3.13: Pile anchor [6].

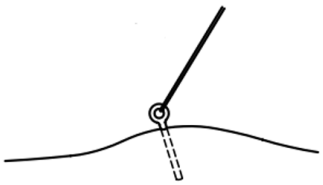


Figure 3.14: Bolt anchor [6].

Some typical anchors used for mooring systems are block, drag, pile and bolt anchors. **Block anchors** are the most common to use concrete. These can be dimensioned up to weigh several tonnes. Block anchors depend on good friction between the ground and the block. **Drag anchors** work by being dragged into the seabed. Whith good friction, the anchor can hold up to 25 times its weight with good seabed conditions. **Pile anchors** can be used when the seabed consists of sand and clay, but the forces that the anchor is able to withstand are not very large. Therefore, the anchors must be large enough to have the desired effect, which in turn increases the cost. **Bolt anchors** can be used where there are opportunities for fastening in rock. The most common is to use bolt anchors for anchoring to land, but there are also opportunities to use such anchors under water [6].

3.2 Hydrodynamics and important parameters

In 2018 a study was conducted to better the understanding of the hydrodynamics of Nylon (PE) net cages. The study investigated the hydrodynamic characteristics of nylon net cages under different attack angles and current flow velocities. The conclusion of the study was that the solidity *(Se)* and the Reynolds number (*Re*) have a significant effect on hydrodynamic coefficients [11]. The **Reynolds number** is defined by

$$Re = \frac{U d_w}{\nu}. \tag{3.1}$$

Here U is the characteristic free-current velocity, d_w is the characteristic length of the body, which in this case is the twine diameter and ν is the kinematic viscosity coefficient. These are the main factors that influence flow [12]. The Reynolds number of different typical aquaculture nets can range between 100 and 10000. The **solidity** on the other hand is defined by

$$Sn = \frac{d_w(2L - d_w)}{L^2} \tag{3.2}$$

and describes the ratio between the projected area of the net over the total area enclosed by the net [13]. Here d_w describes the twine diameter and L represents half the mesh size. Another important parameter is the **wake effect**. This effect is covered in the next section.

3.3 Hydrodynamic models

Scientists have been trying to predict force distribution on aquaculture nets by considerable amounts of experiments and theoretical analyses. Based on the results by these studies hydrodynamic models have been proposed. In general there are two types of hydrodynamic models. That is the **Morison model** and the **Screen force model** [14]. The Morison model treats the forces on the net panel as the sum of the forces on the individual twines, while the screen force model calculates forces by considering the net as a panel [14]. The hydrodynamic forces on a net panel in an oscillatory flow can be defined as

$$\boldsymbol{F} = \rho\nabla\frac{\partial \boldsymbol{u}}{\partial t} + C_a\rho\nabla\frac{\partial \boldsymbol{u}}{\partial t} - C_a\rho\nabla\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{2}C_d\rho A|\boldsymbol{u}-\boldsymbol{v}|(\boldsymbol{u}-\boldsymbol{v}) \tag{3.3}$$

in this equation ρ is the fluid density, ∇ is the submerged volume of the net, A is the projected are of a net panel, u is the fluid velocity vector and v is the structure velocity vector [14]. This equations also contains C_a and C_d which are the added mass coefficient and the drag coefficient. These coefficients are based on experimental data and depends on the Reynolds number, the twine's surface, the Keulegan-Carpenter number, the roughness and the solidity of the net [14]. In equation 3.3

$\rho\nabla\frac{\partial \boldsymbol{u}}{\partial t}$ is the **Froud Krylov force**,

$C_a\rho\nabla\frac{\partial \boldsymbol{u}}{\partial t}$ is the **diffraction force**

$C_a\rho\nabla\frac{\partial \boldsymbol{v}}{\partial t}$ is the **radiation force**,

$\frac{1}{2}C_d\rho A|\boldsymbol{u}-\boldsymbol{v}|(\boldsymbol{u}-\boldsymbol{v})$ is the **viscous force**.

The inertial force is the sum of the **Froud Krylov force**, the **diffraction force** and the **radiation force**. That means the hydrodynamic forces on a net cage is the sum of the inertial force and the viscous force. During transition to steady-state or when the facillity experiences wave loads the inertial force applies [14]. In fact the inertial forces are so small compared to the viscous forces that it's reasonable to ignore the inertial force when dealing with hydrodynamic forces for an aquaculture net cage facility [14].

3.3.1 Morison model

In the Morison force model the twines are regarded as cylindrical elements, while the knots are considered spheres. Calculations using the Morison force model are based on these twines and knots [14]. The viscous force mentioned earlier

$\frac{1}{2}C_d\rho A|\boldsymbol{u}-\boldsymbol{v}|(\boldsymbol{u}-\boldsymbol{v})$ can be decomposed into two components

$\boldsymbol{F_n} = \frac{1}{2}C_n\rho L d_w \left|\boldsymbol{u_n^r}\right| \boldsymbol{u_n^r}$, which is the normal drag force and

$\boldsymbol{F}_t = \frac{1}{2}C_t\rho L d_w \left|\boldsymbol{u}_t^r\right| \boldsymbol{u}_t^r$, which is the tangential drag force.

In these formulas L represents the twine length, d_w describes the twine diameter and ρ is the density of the fluid. The formula for the normal drag force contains $\boldsymbol{u}_n^r$. This is the normal tangential velocity of the fluid relative to the twine. While $\boldsymbol{u}_t^r$ is the tangential velocity of the fluid relative to the twine [14]. C_n is the normal drag coefficient and C_t is the tangential drag coefficient. For numerical simulations the normal and tangential drag coefficients are crucial for force predictions because they determine the amount of force generated. Normally these coefficients are functions of the Reynolds number [14]. The Morison model is directly compatible with the structural model because of its adaptability to the line type elements of the models. This simplifies the implementation of the Morison model for numerical simulation and makes it easier to calculate hydrodynamic forces [14].

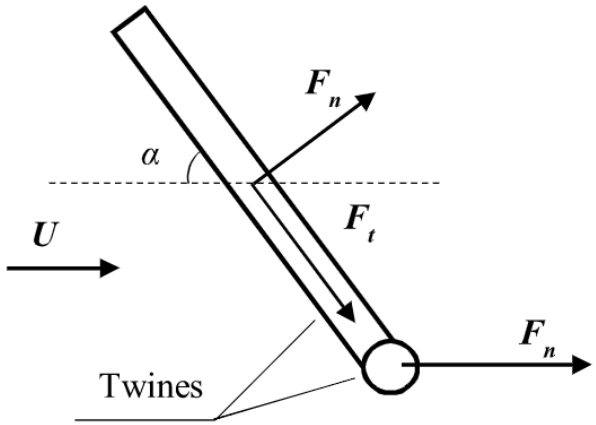


Figure 3.15: Morison model [14].

3.3.2 Screen force model

In the screen force model the net is split into panels which is the basis for calculating the hydrodynamic forces. Instead of the twines and knots being regarded as cylindrical elements like in the Morison model, the twines and knots are instead considered as an integrated structure [14]. The screen force model splits the viscous forces on the net panel. The forces are either split into a force relative to the panel or a force relative to the flow. For some screen force models the viscous force is split into a normal and tangential drag force much like the Morison model. Here the drag forces are related to the orientation of the net panel. This changes the reference area compared to the formula from the Morison model. Instead of having the area being that of the net twine, the area is instead the total are of net panel A_t [14].

$\boldsymbol{F_n} = \frac{1}{2} C_n \rho A_t \left|\boldsymbol{u_n^r}\right| \boldsymbol{u_n^r}$, is the normal drag force and

$\boldsymbol{F_t} = \frac{1}{2} C_t \rho A_t \left|\boldsymbol{u_t^r}\right| \boldsymbol{u_t^r}$, which is the tangential drag force.

In these formulas like the Morison model u_n^r is the normal tangential velocity of the fluid but relative to the net panel instead of the twine. The same applies to $\boldsymbol{u_t^r}$ which is the tangential velocity of the fluid relative to the net panel instead of the twine. C_n is the normal drag coefficient and C_t is the tangential drag coefficient. For the screen force model the normal and tangential drag coefficients of the net panel depends on both the Reynolds number and the solidity. In the Morison model the effect of solidity is not included for the drag coefficients. This is because of the large ratio of mesh size to twine diameter [14].

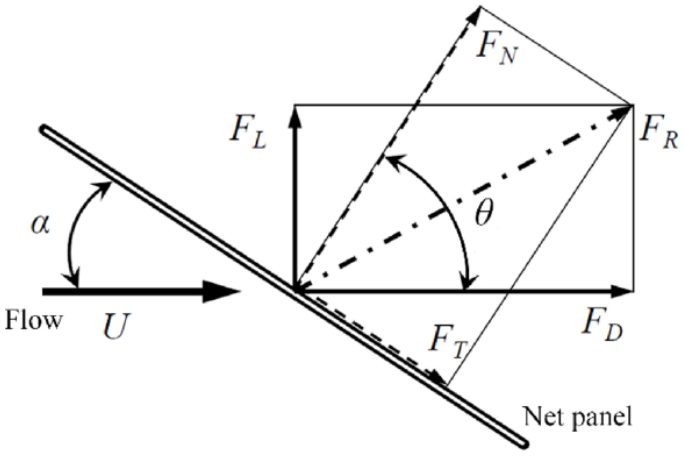


Figure 3.16: Screen force model [14].

The screen force model can also split the viscous force of the net panel into drag and lift forces. These drag and lift forces is relative to the ambient current velocity [14].

$\boldsymbol{F_D} = \frac{1}{2} C_D \rho A_t \left|\boldsymbol{u}^r\right|^2 \mathbf{i_D}$, which is the drag force and

$\boldsymbol{F_L} = \frac{1}{2} C_L \rho A_t \left|\boldsymbol{u}^r\right|^2 \boldsymbol{i_L}$, which is the lift force.

In these formulas A_t is the area of the net panel element, $\boldsymbol{u}^r$ is the fluid velocity relative to the net panel, while $\boldsymbol{i}_D$ and $\boldsymbol{i}_L$ are unit force vectors who indicate the direction of the drag and lift forces. C_D is the drag force coefficient and C_L is the lift coefficient which are both based on experimental data and is dependent on the Reynolds number, solidity and inflow angle [14]. The relationship between $\boldsymbol{C}_D$, $\boldsymbol{C}_L$ and $\boldsymbol{C}_N$, $\boldsymbol{C}_T$ is

$$\boldsymbol{C}_D = \boldsymbol{C}_N \cos\theta \cos^2\theta + \boldsymbol{C}_T \sin\theta \sin^2\theta, \quad \text{and} \quad \boldsymbol{C}_L = \boldsymbol{C}_N \sin\theta \cos^2\theta - \boldsymbol{C}_T \cos\theta \sin^2\theta.$$

3.4 Wake effect

For permeable structures such as net cages of a fish farms the wake effect is an essential and complex mechanism to analyze. Wake is caused by the viscosity of the fluid and creates a region of disturbed flow downstream a structure. For fish farm net cage the wake effect represents the presence of upstream nets modifying the incoming flow velocity for downstream nets [15]. To be able to get a precise force prediction it's imperative to consider the variations of the flow velocity. The square of the flow velocity is proportional to the hydrodynamic force action on the twine [15]. Usually when considering fish farm net cages and the wake effect, there are three types of wake that is considered. Twine-to-twine, net-to-net and cage-to-cage. In this thesis only one cage is considered. That means only twine-to-twine and net-to-net wake effect is relevant.

3.4.1 Twine-to-twine wake effect

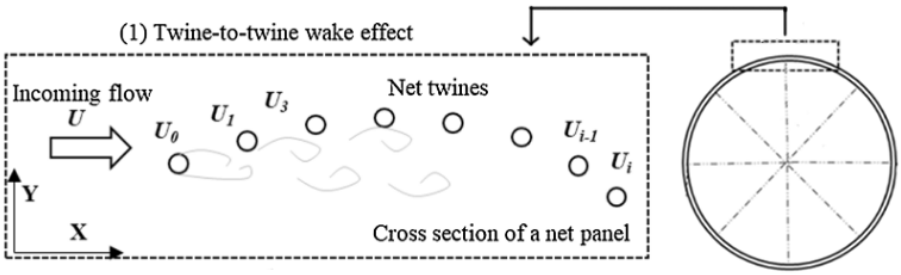


Figure 3.17: Twine-to-twine wake effect [14].

The twine-to-twine wake effects is the result of the interactions between the net twines. When the inflow angle of the net plane is larger than 70° the velocity of the downstream twine will be smaller than that of the upstream twine [14]. When using the twine-to-twine wake effect with the Morison model a function describing the flow pattern behind a cylinder needs to be made. This can be done by using Blevins formula [14].

$$\boldsymbol{U}_{\text{downstream}} = \boldsymbol{U}_{\text{upstream}} \left(1 - 1.02 \sqrt{\frac{C_d}{6 + x/d_w}} \exp\left\{ \frac{-(y/d_w)^2}{0.0767 C_d \, (6 + x/d_w)} \right\} \right)$$

Here $U_{\text{downstream}}$ represents the velocity of the downstream cylinder at (x,y). The wake effects experienced by a single twine in the Morison models depends on contributions from all the other twines in the model [14]. The screen force model automatically includes the twine-to-twine wake effect because of the hydrodynamic coefficients of the net panels considering the interactions between the twines implicitly [14].

3.4.2 Net-to-net wake effect

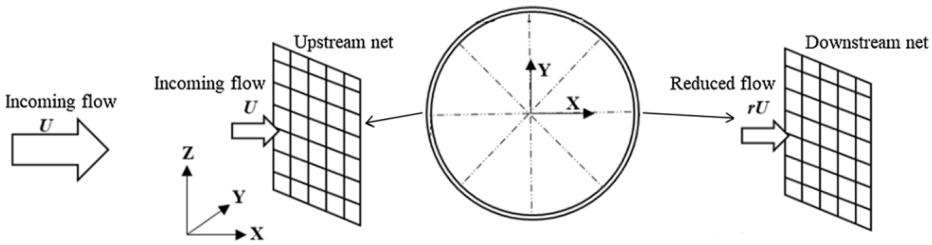


Figure 3.18: Net-to-net wake effect [14].

The net-to-net wake effect is the result of the interaction between nets inside a single fish farm net cage. The net-to-net wake effect that is experienced by a single cylindrical net cage is approximately half. If the net-to-net wake effect is neglected the forces on the mooring system can be overestimated up to 22% [16]. For numerical simulation a flow reduction factor r is used to represent the net-to-net wake effect. This flow reduction factor usually ranges from 0 to 1 [14].

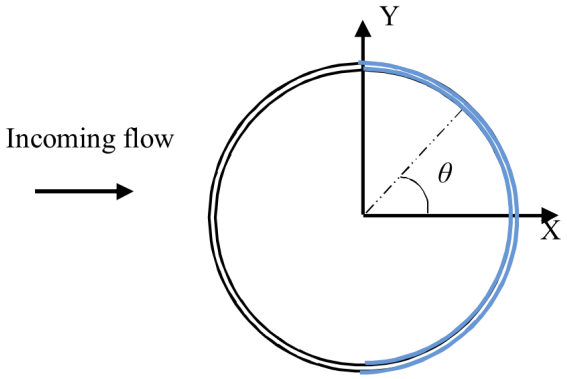


Figure 3.19: Net experiencing wake [14].

3.5 FhSim

FhSim is a software platform and framework for mathematical modelling and numerical simulation. FhSim was developed at Sintef Ocean. The main use of FhSim is for marine applications such as research and as a basis for industrial tools and services. The framework is aimed at simulating nonlinear systems in the time domain. This is done by using models described by ordinary differential equations [17].

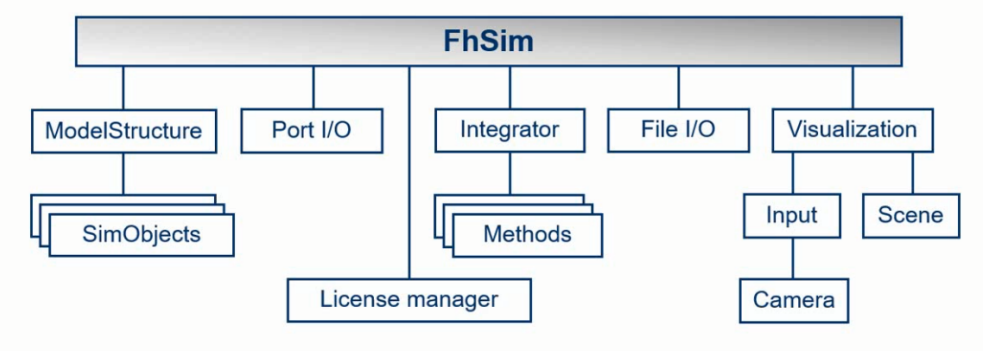


Figure 3.20: FhSim main components framework [17].

SimObjects are objects that implement different sub-models. The communication between these simulation objects is made possible by the input/output ports. The simulation objects are responsible for computing its own state derivatives based on ordinary differential equations [18].

ModelStructure presents a collection SimObjects as a single model to the rest of the framework. This creates an overview of all the SimObjects, their states and how these are interconnected through the input/output ports [18].

Integrator is responsible for keeping track of the total system model states. The integrator advances the total system model states by using the state derivatives presented by the ModelStructure [18].

Input- and output files deals with all the interaction between the model system and the files. The input file is used to setup the simulation and an output file to export the simulation results [18].

3.5.1 Cage collar

FhSim uses the principle of modal superposition and summarizes these to model the structural response of the floating collar [17].

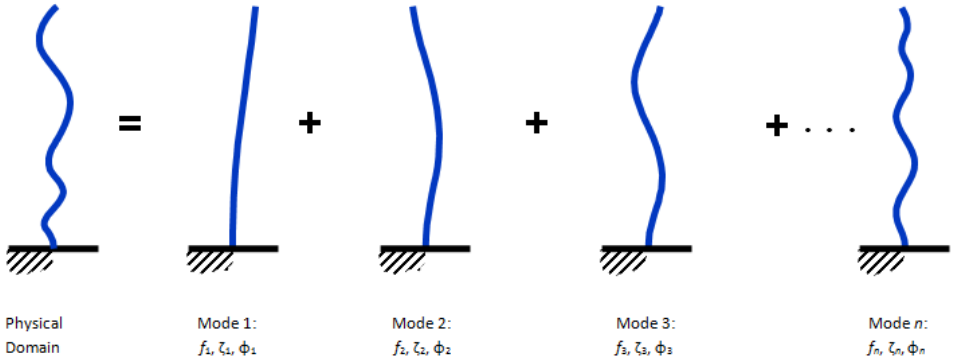


Figure 3.21: Modal superposition [19].

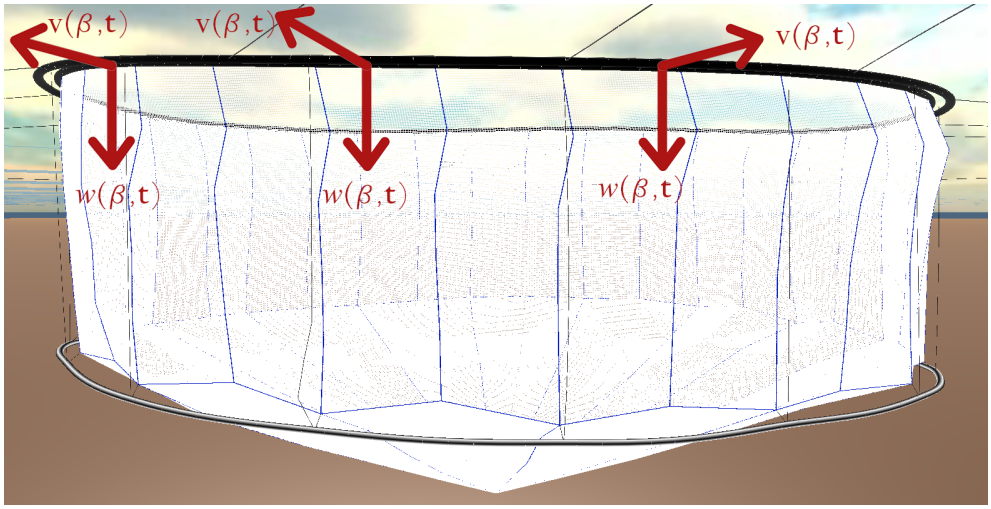


Figure 3.22: FhSim floating collar [17].

Where v is the horizontal displacements at a given point on the cage collar.

$$v(\beta, t) = \sum_{n=2}^{N} (a_n^h(t) \cos n\beta + b_n^h(t) \sin n\beta)$$

and w is the vertical displacements at a given point on the cage collar.

$$w(\beta, t) = a_0^v(t) + \sum_{n=1}^{N} (a_n^v(t) \cos n\beta + b_n^v(t) \sin n\beta)$$

Together the cosine and sine modes give symmetric deformations around the x- and y-axis which are able to produce arbitrary non-symmetric motions.

3.5.2 Net

In FhSim the sea cage net geometry is modelled by a collection of triangular net elements. The structural forces within each triangular element is calculated as a function of the relative motions between the three loads associated with the triangular element. The net deformation is then modelled by the motions of the loads under the effects of both internal structural forces and the external hydrodynamic forces [17].

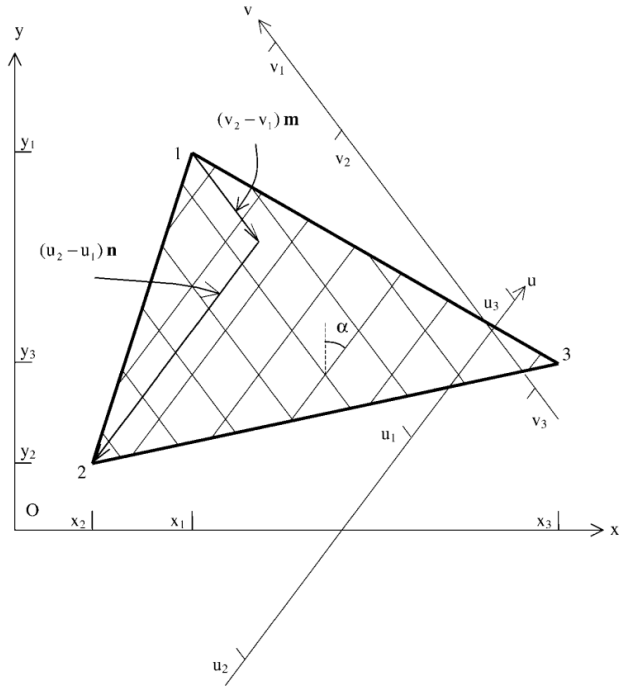


Figure 3.23: Triangular elements for nets [20].

FhSim is then able to calculate the force vector on nodes 1-3 of each triangular element by these equations [20].

$$f_1 = -\frac{C_u}{2}\begin{Bmatrix} (v_3 - v_2)\frac{ny}{n^2} + (u_3 - u_2)\frac{my}{m^2} \\ (v_2 - v_3)\frac{nx}{n^2} + (u_2 - u_3)\frac{mx}{m^2} \\ 0 \end{Bmatrix}$$

$$f_2 = -\frac{C_u}{2}\begin{Bmatrix} (v_1 - v_3)\frac{ny}{n^2} + (u_1 - u_3)\frac{my}{m^2} \\ (v_3 - v_1)\frac{nx}{n^2} + (u_3 - u_1)\frac{mx}{m^2} \\ 0 \end{Bmatrix}$$

$$f_3 = -\frac{C_u}{2}\begin{Bmatrix} (v_2 - v_1)\frac{ny}{n^2} + (u_2 - u_1)\frac{my}{m^2} \\ (v_1 - v_2)\frac{nx}{n^2} + (u_1 - u_2)\frac{mx}{m^2} \\ 0 \end{Bmatrix}$$

FhSim also uses a simplified net model for simulations with fish in the net cage. This simplified model uses a truss element to calculate the structural forces within the net. The structural model is then combined with the Baumgarte stabilization to make the model more computationally efficient [17].

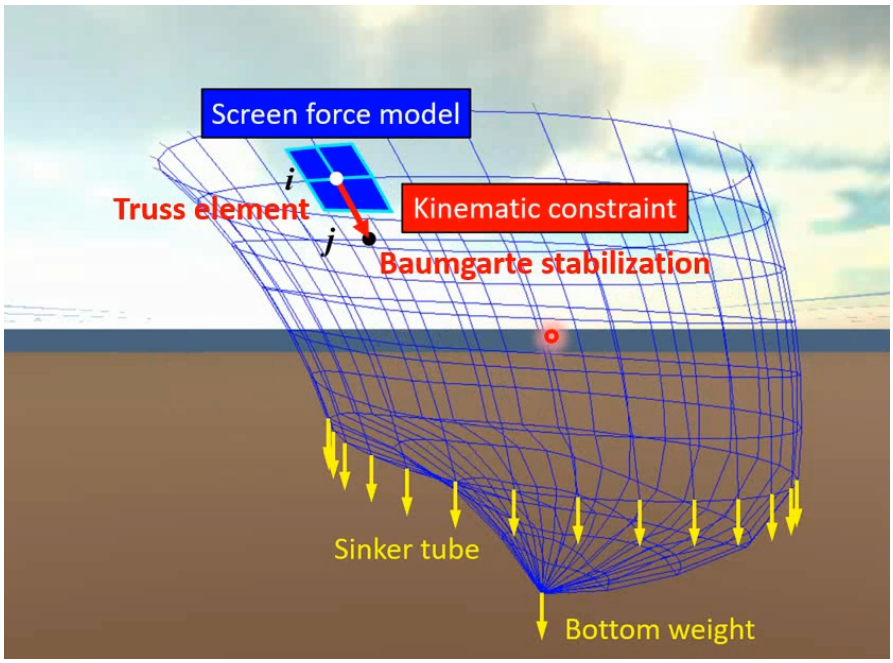


Figure 3.24: FhSim real-time simulation with fish [17].

The screen force model is used instead of the Morison type force model, because it has been found to result in more accurate calculation for the lift F_L and drag F_D forces [21]. When using the screen force method the solidity ratio and Reynolds number are important parameters.

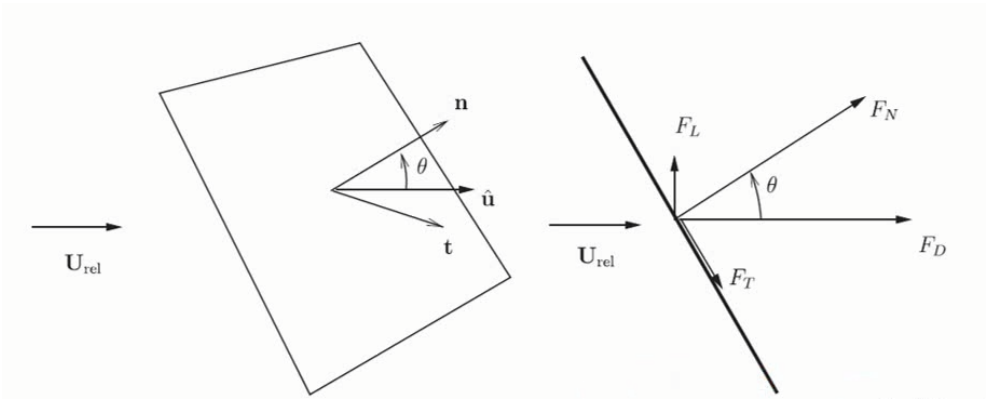


Figure 3.25: Screenforce model [21].

3.5.3 Cables, ropes and chains

FhSim uses a generic cable model to simulate cable, ropes and chains. This generic cable model consists of interconnected rigid bar elements with constraint equations. The constraint equations are used to calculate the structural forces like the tension, bending and torsion connected to the rigid bar elements. Like the net, FhSim also combines the structural model with the Baumgarte stabilization to make the model more computationally efficient [17].

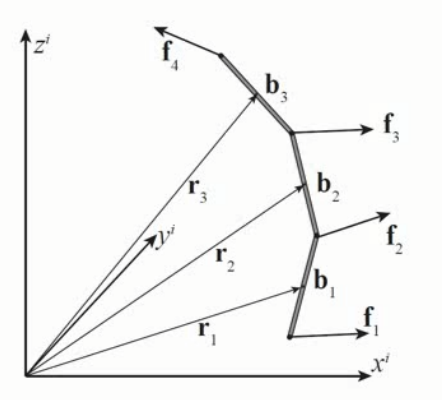


Figure 3.26: Interconnected model [17].

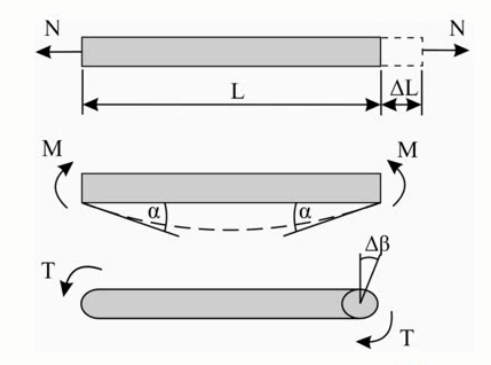


Figure 3.27: Rigid bar element [17].

The force model used for calculating the hydrodynamic forces is the **Morison** type force model. The Morison type force model is then applied to each rigid bar element. The external forces that affect cable model are the gravity forces, hydrostatic forces, Froude-Kriloff forces, diffraction forces and non-linear viscous forces [17].

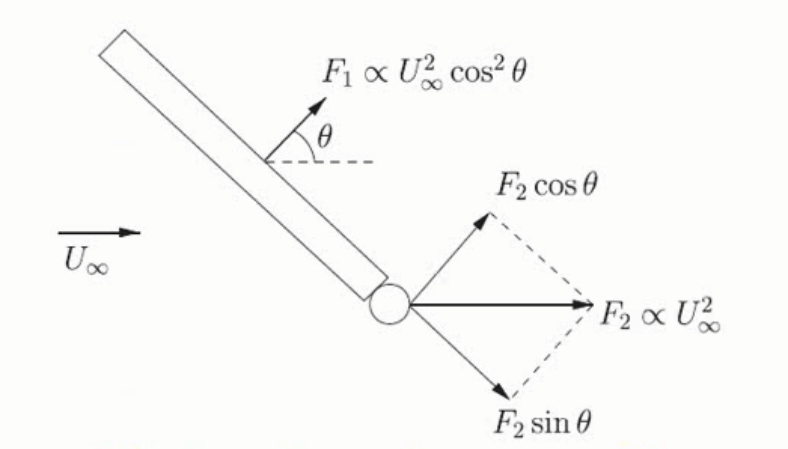


Figure 3.28: Morison force type model [17].

3.5.4 Bottom weight sinker tube

FhSim models the sinker tube as a specific version of the generic cable model where the two endpoints are interconnected to establish a continuous cable structure [17].

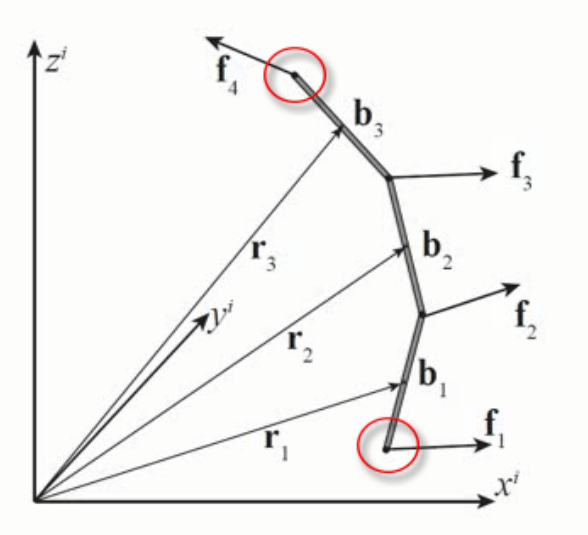


Figure 3.29: Interconnected endpoints [17].

The sinker tube is affected by the same external forces as the cable model and also forces exerted by other sub-models attached to the sinker tube. These external forces are found separately for each element that affects the sinkertube. For the sinkertube the generic cable model also uses the constraint equations to make sure the element motion results in the correct structural response with respect to the structural properties of the sinker tube [17].

Figure 3.30: Sinkertube [22].

3.5.5 Buoy

Buoys in FhSim is modelled as vertical circular cylinders. The buoys are also modelled with a conical bottom. The buoy is affected by the same forces as the generic cable model. The excitation forces on the buoys and the acceleration between buoy and water particles are calculated by using linear long wave theory. The hydrodynamic forces will affect the mean wetted surface of the buoy. The mean wetted surface of the buoy is the bottom cone part where FhSim calculates the hydrodynamic forces [17].

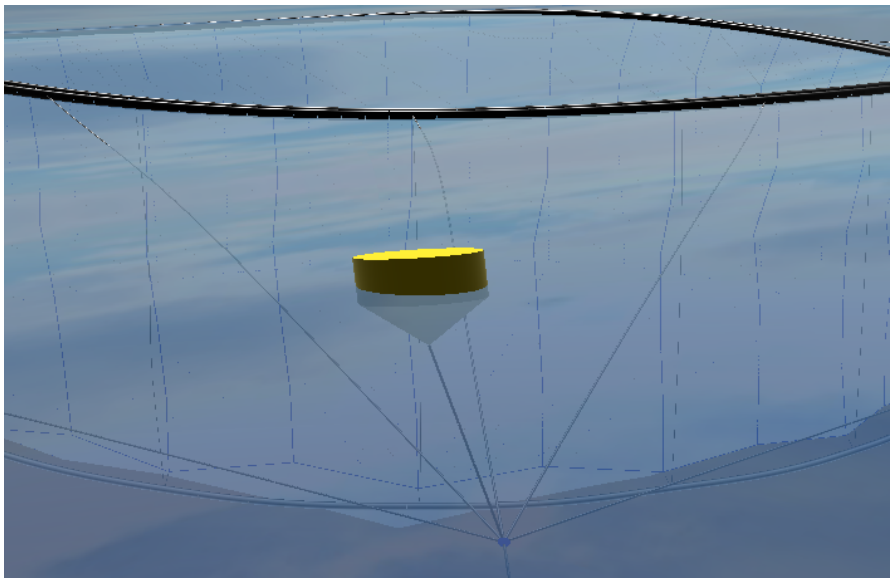


Figure 3.31: FhSim buoy [17].

3.5.6 Output data

Sintef has conducted research with FhSim to see if the software generates data that produces the same results as real experiments. In a validation research paper from 2014 [23], Sintef concludes that FhSim produces realistic output data.

3.6 Machine learning

Machine learning is a form of artificial intelligence where statistical methods are used with computers to find patterns in large amounts of data. That way, the machine learns instead of being programmed [24]. In order for a machine to be able to learn, data is needed. Such learning is also called training. The data sets used for such training are usually divided into a training set and a testing set. The training set is used to train the model. To check if the machine has learned something from the training set, the testing set is used. The testing set will then be data that the machine has not seen before and this makes it possible to see what the model has learned. Within machine learning, there are three main categories. These are supervised learning, unsupervised learning and reinforcement learning [24].

Supervised learning is a type of machine learning where the machine is trained to predict the output values based on the input values. The machine does this by finding a function between the input value and the output value [24].

Unsupervised learning is a type of machine learning where the machine lacks the correct output data from multiple input data. With unsupervised learning, the algorithm rather tries to find structure in the input data. The algorithm can do this by grouping them in clusters [24].

Reinforcement learning is a type of machine learning where the model constantly interacts with an environment. This environment provides continuous punishment or reward that affects the learning. That is, continuous correction that leads to reinforced desired behavior without specifying how it should be done [24].

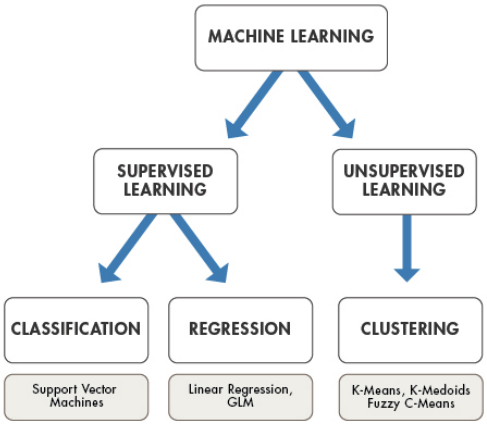


Figure 3.32: Machine learning types [25].

3.6.1 Machine learning regression

To understand the relationship between and independent variable or feature, and a dependent variable or outcome, the regression method can be used. The outcome can be predicted once the relationship between the independent and dependent variables have been estimated. The regression method is used to predict outcomes from data. This is because the method works as a continuous outcome forecasting method. The regression method is a key element of predictive modeling, thus it can be used for many different applications of machine learning [26].

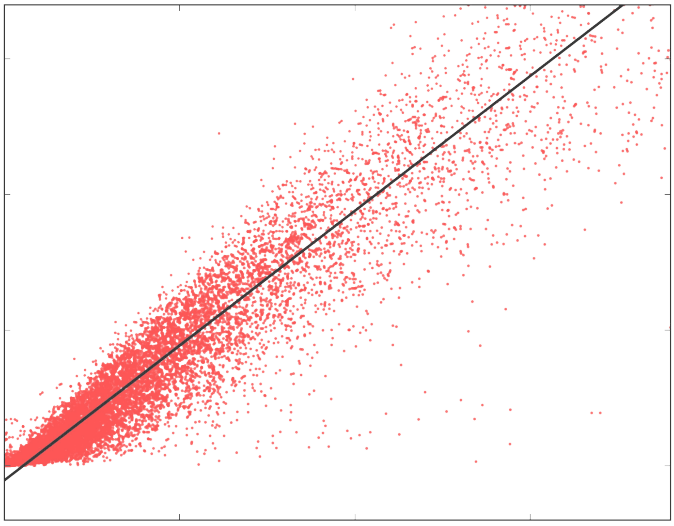


Figure 3.33: Linear regression [27].

3.6.2 Gradient boosting

Gradient boosting is the process of creating a prediction model based on an ensemble of weak prediction models. Gradient boosting can be split into three elements. That is loss function to be optimized, a weak learner to make predictions and additive model to add weak learners to minimize the loss function [28]. The **loss function** works as a indicator for the distance between the present output of the algorithm and the expected output. For regression the loss function is classified as continuous values [29]. The loss function must be differentiable and depends on the problem type being solved [28]. In gradient boosting, decision trees are used as the **weak learner**. These decision trees are regression trees that are used to correct the residuals in the predictions by outputting real values for splits that can be added together. The regression trees are greedy and choose the best split points based on purity scores. The weak learners are commonly constrained in different ways. This can be a limit on the maximum number of layers, nodes or splits [28].

The **additive model** adds one tree at a time while the existing trees in the model doesn't change. When the model is adding the trees it's important to minimize the loss. This is done by using a gradient descent procedure. A gradient descent procedure is performed by minimizing parameters so that it's possible to calculate the error or loss and update those parameters. For gradient boosting, instead of parameters there are weak learner sub models. These sub models are the decision trees mentioned earlier. After calculating the loss, the gradient descent procedure is performed by adding a tree to the model so that the loss is reduced. The loss is reduced by parameterizing the tree and then modify the parameters of the tree and reducing the residual loss [28].

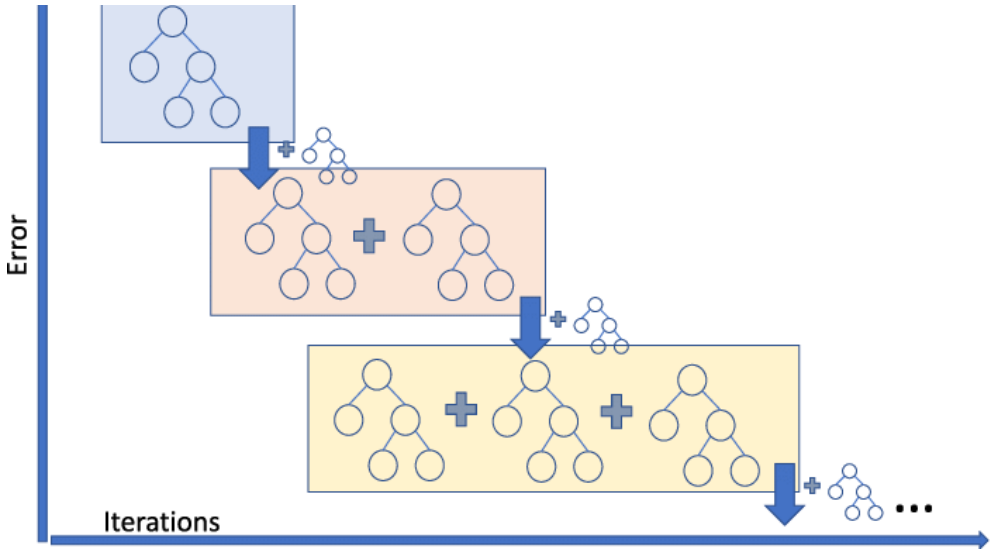


Figure 3.34: Gradient boosting [30].

3.6.3 XGBoost algorithm

XGBoost is a form of supervised learning algorithm. This is because XGBoost uses gradient boosting which is a way of accurately attempting to predict a target variable by combining estimates from simpler and weaker models. XGBoost is a implementation of the gradient boosted tree algorithm [31].

Like mentioned earlier the weak learners in gradient boosting is the regression trees. The regression trees have leafs. These leafs is made by the regression trees mapping an input point that contains a continuous score [31]. XGBoost works by minimizing a regularized L1 and L2 objective function. This objective function combines a convex loss function and a penalty term for the model complexity. For XGBoost the convex loss function is based on the difference between the predicted and target outputs. The penalty term is the regression tree functions [31]. This procedure repeats by adding new trees that predict the residuals or errors of previous trees. These errors are then combined with previous trees to make the final prediction [31].

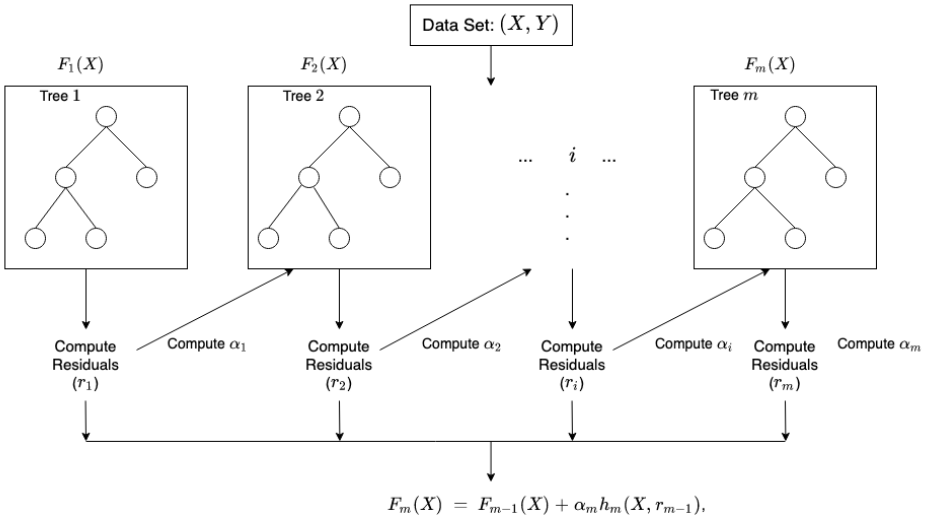


Figure 3.35: XGBoost [31].

in figure 3.35 $\boldsymbol{\alpha_i}$ is the regularization parameters and $\boldsymbol{r}_i$ are the residuals computed with the $\boldsymbol{i^{th}}$ tree. Where $\boldsymbol{h_i}$ represents a function that is trained to predict residuals $\boldsymbol{r}_i$ using X for the $\boldsymbol{i^{th}}$ tree. [31]. The residuals computed $\boldsymbol{r}_i$ are used to compute $\boldsymbol{\alpha_i}$. Then it's possible to compute

$$\arg\min_{\alpha} = \sum_{i=1}^{m} L\left(Y_i, F_{i-1}\left(X_i\right) + \alpha h_i\left(X_i, r_{i-1}\right)\right)$$

where $L(Y, F(X))$ is a differentiable loss function [31].

3.7 AKVA Group oceanographic environmental buoy

AKVA's oceanographic environmental buoy is an electric buoy that monitors important parameters from the environment. The buoy can measure current velocity and its direction at three different depths. It also measures wave height and wave direction. The buoy is made of polyethylene (PE) material and has a net buoyancy of 500kg. The buoy is composed of a number of different components. These are buoyancy body, mast, navigation lights, battery, sensors and steel luminaire for mounting the mast and mooring [32].

The buoy measures the current and the current direction with a current profile Doppler sensor. The buoy is placed outside the facility and is not used by boats and other vessels. The battery life of the battery is around two years depending on temperature and logging interval. The height of the buoy is 125cm, the diameter is 119cm and the weight of the buoy is 70kg. While the buoy is usually placed close to the fish farm facility the communication range of the buoy is 1.5 km.

Figure 3.36: AKVA Group buoy detail [33].

3.7.1 AKVA Group buoy setup

In order for the buoy to function optimally, a stabilization weight is usually fitted under the buoy. This is usually 100kg chain 10 meters below the buoy. In order to prevent corrosion, the 10 meter long rope that is connected between the buoy and the chain is assembled with a stainless steel casing and shackle [32].

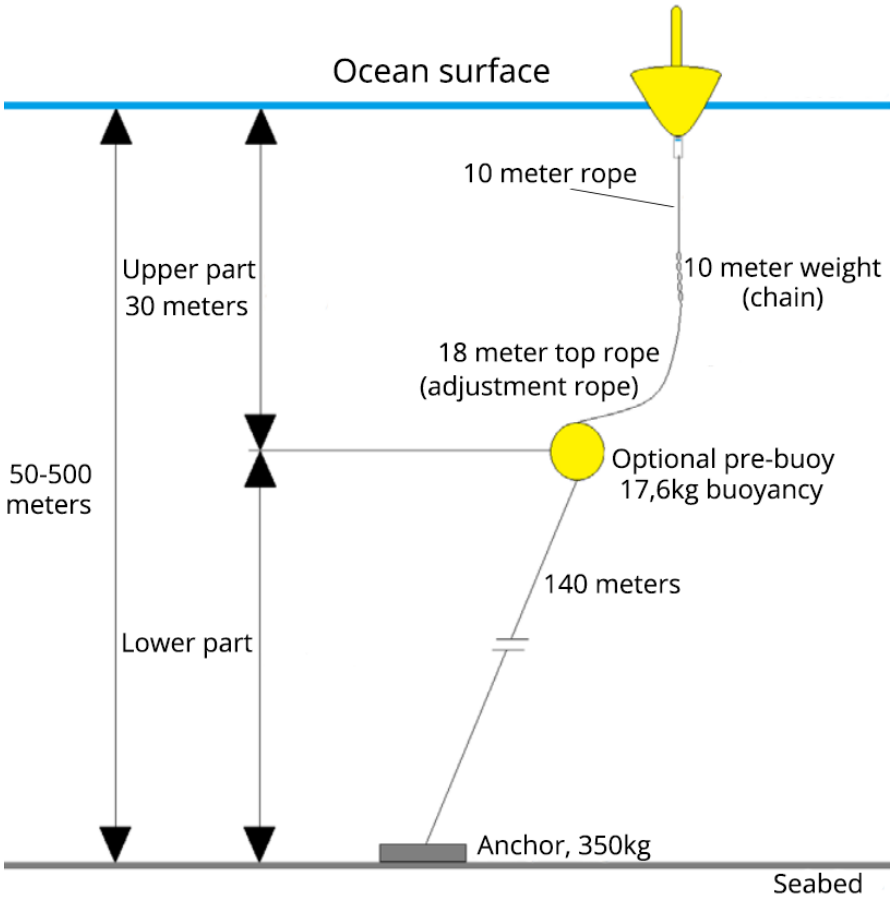


Figure 3.37: AKVA buoy setup [32].

3.7.2 Cloud based data storage

The buoy logs measurements digitally up to a cloud storage system. Data on waves, current, temperature and salinity are stored here. The storage interval is usually set to 10 minutes to reduce the drainage of the battery. Data stored in the cloud system is displayed as shown in Figure 3.30. The storage data is available for download from any time interval in both excel and JSON format.

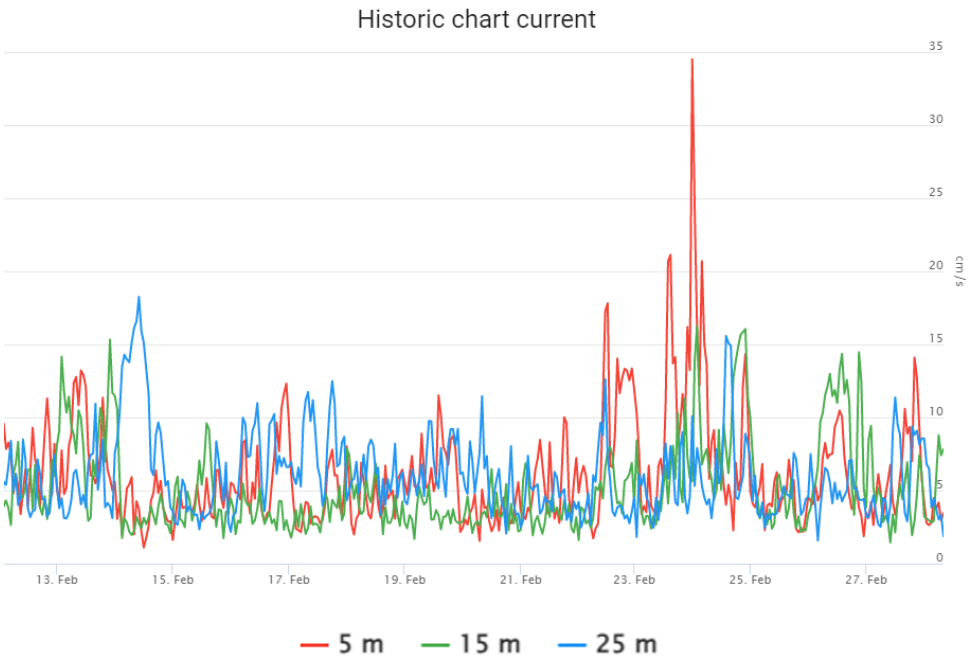


Figure 3.38: Historic current chart.

Chapter 4

CAC case study

This chapter presents the case study for this thesis. AKVA Group only has one location with a sensor buoy that measures current, waves, salinity and temperature. That is at the CAC project at Vindsvik in Hjelmeland. CAC stands for Center for Aquaculture Competence. This is a research fish farm facility that contains seven fish farm net cages. The project is owned by MOWI which are one of the worlds larges companies within fish farming.

Figure 4.1: CAC project [34].

4.1 CAC Overview

The CAC project is a fish farm composed of several cages but for this case study, the simulations have been reduced to one of the cages at the CAC facility. This is to reduce the necessary computing power required and to increase the amount of simulations that can be performed. The chosen net cage that is used for simulations in this thesis is cage 1 shown in figure 4.2 and 4.3.

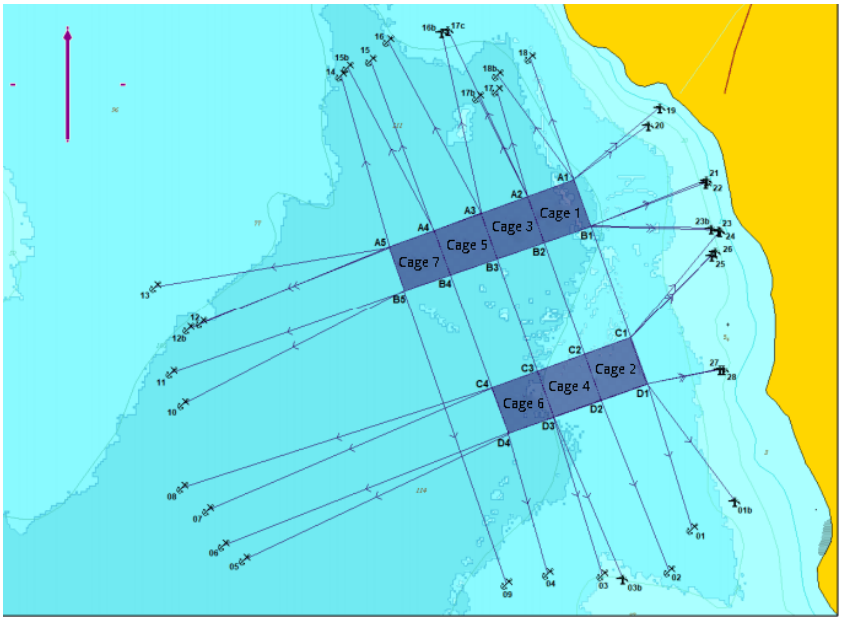


Figure 4.2: CAC Overview.

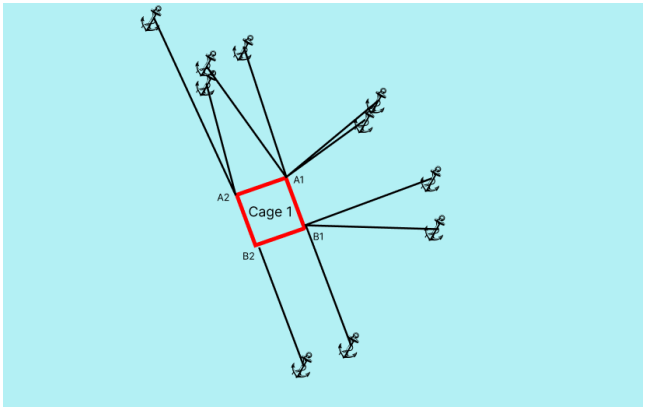


Figure 4.3: CAC Cage 1.

4.2 CAC Net cage

The CAC facility uses a type of net cages called angled circle net cage, which can also be referred to as a Spaghetti net cage type. The reason it's called a spaghetti net cage is because of the extra lines at the bottom of the cage. The Spaghetti cages on the CAC project is shown in Figures 4.4, 4.5, 4.6 and 4.8.

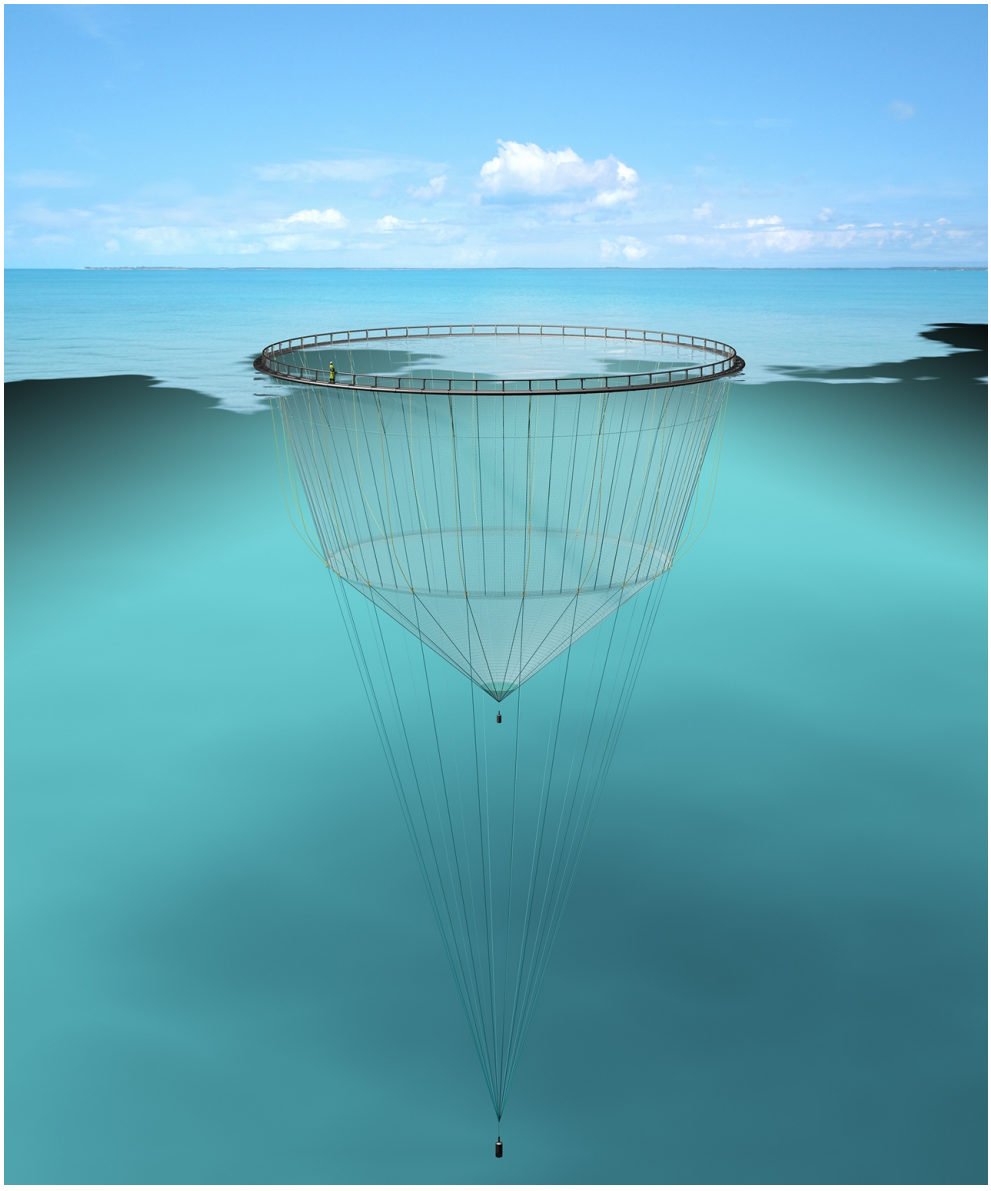


Figure 4.4: CAC angled circle net cage [35].

4.2.1 Spaghetti net cage design

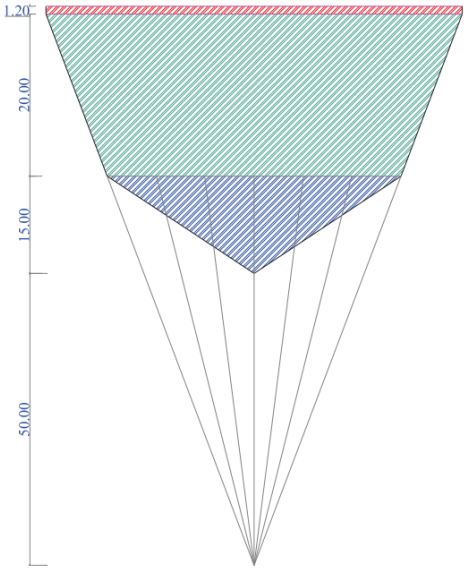


Figure 4.5: CAC Spaghetti cage detail.

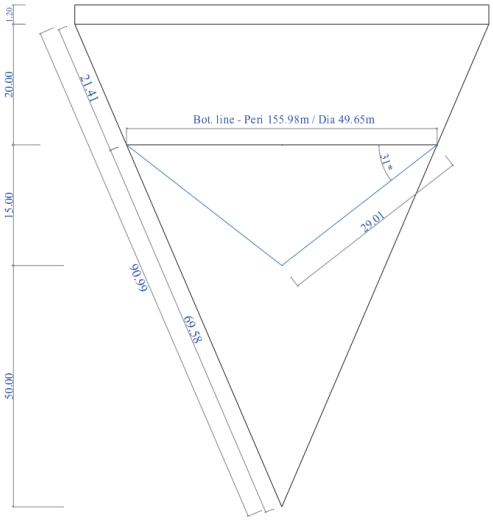


Figure 4.6: CAC Crosssectional view.

Location	Depth	Surface Area	Volume
	(Mtr)	(Square Mtr)	(Cubic Mtr)
	1.20	244	3,972
	20.00	3,857	51,881
	15.00	2,266	9,684
	50.00	Only Ropes	Only Ropes

Figure 4.7: CAC Spaghetti cage detail table.

The spaghetti net cage is a knotted cage and is made of nylon. The net cage has not been treated. The total depth of spaghetti net cage without the spaghetti ropes are 35 m but the total depth including the spaghetti is 85 m. The thread thickness of the net bag is 2.5 mm and the solidity of the net is 19.2. The dimension class of the net cage is 0 but the reliability class is 2. The cage also has a jumping net which has a height of 1.2 m. The bottom rope is 2 kg/m and the square weight of the net bag is 0.35 g/m^2. The depth of the bottom of the net cage is 15 m and the sinkertube that the CAC spaghetti net cage uses is called Aqua Ring. The half mesh size of the spaghetti net cage is 19.5.

4.2.2 CAC Net cage ropes

The ropes used for the CAC net cage system is different types of Danline ropes. The specific rope specifications are shown in Table 4.1 and 4.2. Danline rope is made with a mix of polypropylene and polyetylene [36].

Figure 4.8: Danline [36].

	Top rope	Depth band
Type	Danline rope	Danline rope
Diameter (mm)	22 mm	22 mm
Lenght (m)	204 m	21.2 m
Loops (quantity)	72 qt	72 qt
Lenght loops (cm)		30 cm
Felling thread	2.5 mm	2.5 mm
Treatment felling thread	White	White

Table 4.1: CAC Net cage ropes 1.

	Main rope	Bottom rope	Cross rope
Type	Danline rope	Danline rope	Danline rope
Diameter (mm)	22 mm	22 mm	22 mm
Lenght (m)	204 m		0
Loops (quantity)	72 qt		12 qt
Lenght loops (cm)	15 cm		
Felling thread	2.5 mm		
Treatment felling thread	White		

Table 4.2: CAC Net cage ropes 2.

4.2.3 CAC Floating Collar

The CAC fish farm is using a floating collar made by Aqualine. The name of the collar is FrøyaRingen FR500-200G and is made with polyethylene (PE). FrøyaRingen consists of two floating collars. The floating collars circumference is 200mm while the diameter of the collars are 500mm. The floating collar has a net buoyancy of 49.2 tons and the solidity is 30.9%. The assumed service life of the floating collar is 20 years. The floating collar is designed and documented to withstand a maximum current speed of 1 m/s. The weight of the collar is 80 kg/m which totals to 16.5 tons.

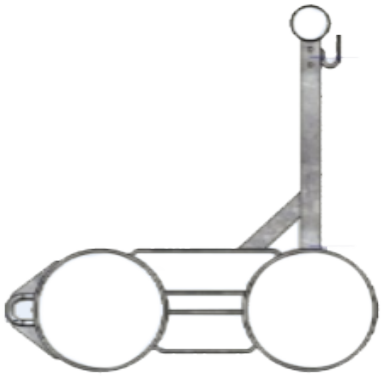


Figure 4.9: CAC Floating collar 1.

The maximum bending radius of the diameter is 15% and the hen foot moorings maximum angle is 35-50° with an opening of 2x14.4 m.

Figure 4.10: CAC Floating collar 2.

4.2.4 AKVA Group buoy sensor placement CAC

MOWI is a customer of AKVA Group and currently have one of the AKVA Group sensor buoys at the CAC facility. The AKVA Group sensor buoy is placed close to cage 7. The buoy was first activated in July 2021.

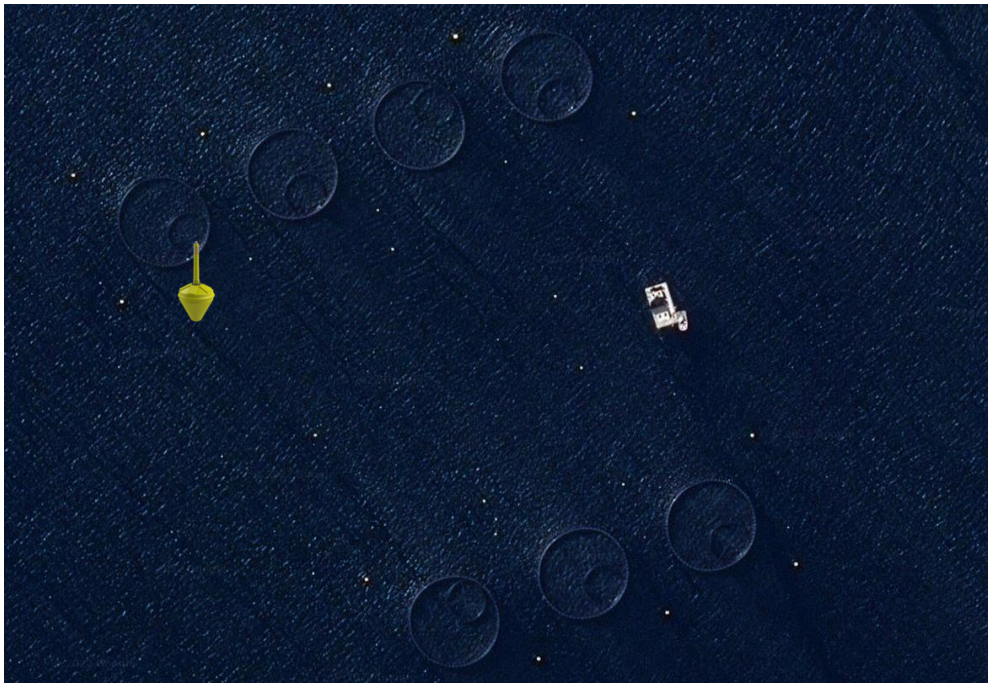


Figure 4.11: AKVA Group sensory buoy placement.

Chapter 5

Numerical simulation

This chapter goes through the setup for the simulations conducted in FhSim. The chapter shows the various components in FhSim and which of these who have been chosen for use as further results for machine learning. An explanation of the forces acting on the chosen components are also included. Furthermore, the simulation setup for the CAC project case study is included. Here elaborate details of the design of the net cage system on CAC are shown. This includes the floating collar, net structure, sinkertube, buoys, mooring line, frame cable and bridles.

The chapter also shows which current directions that was used for the simulations in this thesis. The current velocities used for these different directions are also included here. Additionally, there is a review of why the results for machine learning has been extracted as a result from the simulations with an average from 500s to 600s. The process of completing the 640 simulations is also included.

Finally, the chapter goes through the process of treating the output data. Here, the process around clearing the large result files is explored. This includes getting rid of unnecessary data, calculation of the resultant forces and how the end result data that is used for machine learning is stored.

5.1 FhSim General configuration

This is what a single net cage system looks like in FhSim. B1-B12 represents the bridles connected by the hen foot to the floating collar. AC1-AC8 represents the anchor cables and FC1-FC4 represents the frame cables. FhSim is able to calculate the force distribution on these different units in the single net cage system. Each unit has it's own force components.

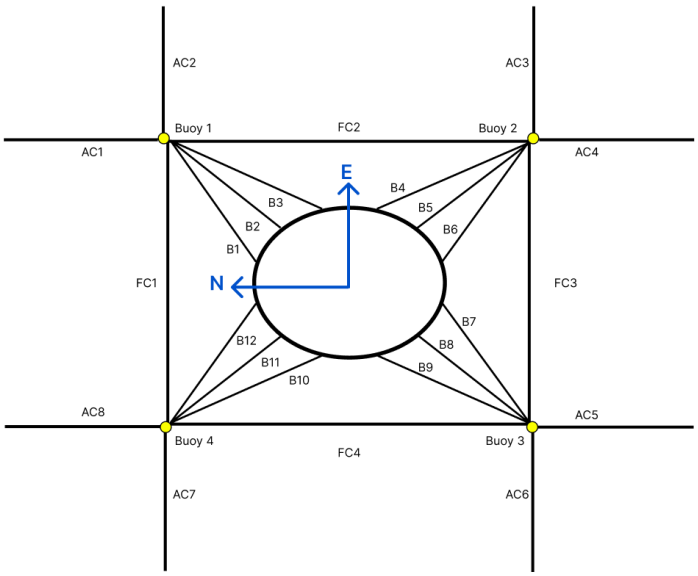


Figure 5.1: FhSim Single net cage system.

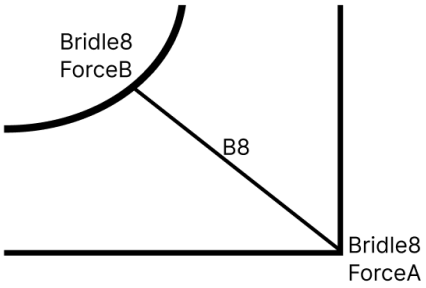


Figure 5.2: FhSim Force components.

Each force component is split into three axes. Force A i split into Force A_0, Force A_1 and Force A_2 where A_0 is the force in north direction, A_1 is the force in east direction and A_2 is the force in the downwards direction. In this master's thesis, the force distribution on the bridles have been selected as the result values to use with the machine learning algorithm. For the Bridle force distribution it's the A force that is used. This is the connection force to the mooring frame.

5.2 FhSim CAC single net cage configuration

The spaghetti cage net type that is used in the CAC project has been modeled in FhSim. The dimensions and properties used in FhSim for the CAC single net cage system is given in table 5.1.

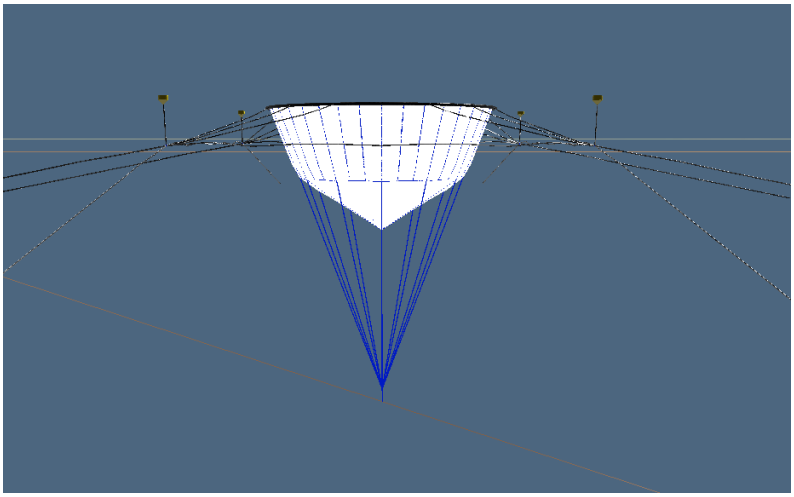


Figure 5.3: FhSim CAC Visualization 1.

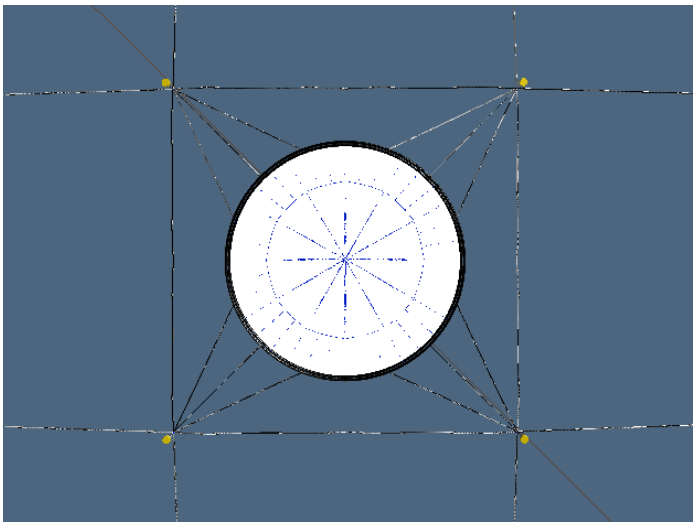


Figure 5.4: FhSim CAC Visualization 2.

Component	Parameter	Value	Unit
Floating Collar	Floater diameter	65.8	mm
	Pipe diameter	50	mm
	Wall thickness	36.8	mm
	Pipe E-modulus	0.9	GPa
	Pipe weight	81	Kg/m
	Drag coefficient	1	—
Net structure	Twine diameter	2.5	mm
	Twine length	19.5	mm
	Density	1125	kg/m^3
	Net E-modulus	1	GPa
Buoy	Diameter	2	m
	Vertical cylinder depth	1	mm
	Conical bottom depth	2	m
Mooring line	Lenght	101.98	m
	Diameter	0.10	m
	Mooring line E-module	0.415	GPa
	Mooring line weight	8.33	Kg/m
Frame cable	Lenght	9.3	m
	Diameter	0.064	m
	Frame cable E-module	1.87	GPa
	Frame cable weight	3.29	Kg/m
Bridle	Length (center)	38,7	m
	Length (sides)	43.65	m
	Diameter	60	mm
	Bridle E-modulus	2.226	GPa
	Bridle weight	3.87	Kg/m

Table 5.1: FhSim CAC net cage system setup.

5.3 FhSim simulation set-up

By studying the sensor data registered by AKVA Groups sensor buoy. The highest registered currents for each depth have been used to create a range of current velocities for the numerical simulations in this thesis. In this thesis 640 different cases have been simulated in FhSim. For these cases the only varying input parameters are the current in the top depth layer as well as the current in the bottom depth layer. The top current ranges from 0 m/s to 0.9 m/s and the bottom current ranges from 0 m/s to 1.5 m/s. The simulation cases are run with the currents flowing in the same direction. This is to reduce the simulation time as much as possible. These 640 cases are split into 160 cases for 4 different current directions. The current directions are 1.57 rad $(1/2\pi)$, 2.09 rad, 2.57 rad and 3.14 rad (π). Other reductions to the complexity of the simulation cases include excluding waves. This is done for the same reason as the current directions. The reason behind choosing these current directions is because of the ability to reflect the bridle forces on to the other parts of the net cage system by using symmetry.

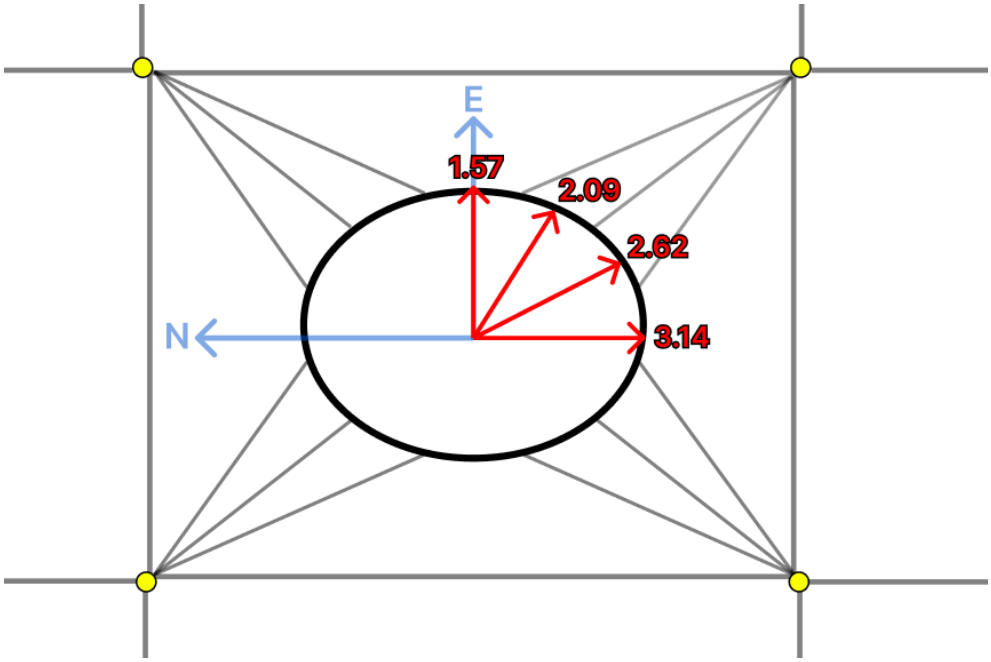


Figure 5.5: FhSim simulation directions.

5.4 FhSim Case examples

Table 5.2 shows an example of different input values for 16 out of 640 simulations. For all the cases in this thesis the only input parameter that changes is the top and bottom current velocity together with the current's direction.

Top Current	Bottom Current	Direction
0.1	0.0	3.14
0.1	0.1	3.14
0.1	0.2	3.14
0.1	0.3	3.14
0.1	0.4	3.14
0.1	0.5	3.14
0.1	0.6	3.14
0.1	0.7	3.14
0.1	0.8	3.14
0.1	0.9	3.14
0.1	1.0	3.14
0.1	1.1	3.14
0.1	1.2	3.14
0.1	1.3	3.14
0.1	1.4	3.14
0.1	1.5	3.14

Table 5.2: FhSim Input value example.

5.5 Simulation duration

All the simulations in this thesis have been simulated from 0 s to 1000 s. Though only the results from 500 s to 600 s is used for machine learning. This is because the model needs to achieve steady state condition. Steady state condition means that the force distribution of the bridles together with the vertical position of the bottom net has reached its highest value for that specific case. This is the important values to use for machine learning to predict the force distribution for the net cage model. Between 500 s and 600 s the model is guaranteed to reach steady state for these simulation cases in this thesis.

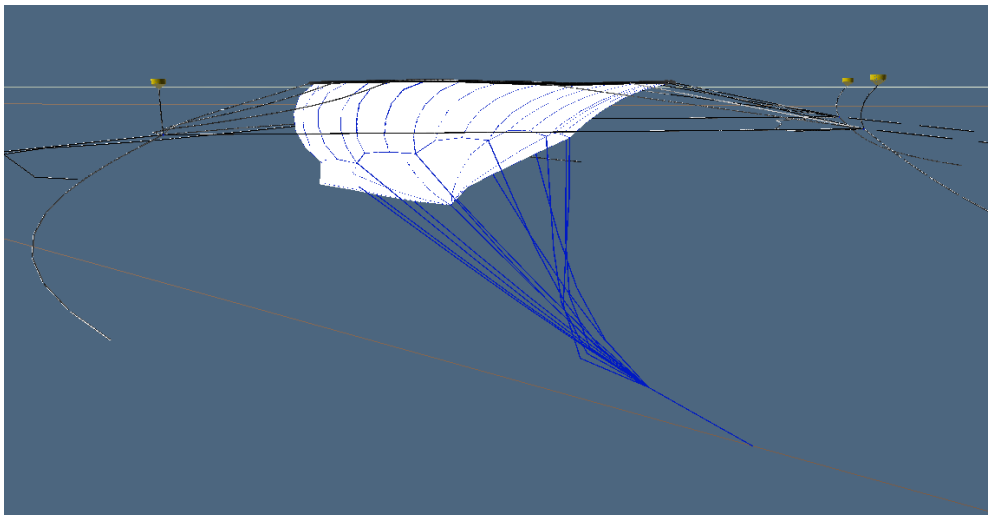


Figure 5.6: FhSim steady state 1.

Figure 5.7 shows the Bridle ForceA sum for top current and bottom current 0.2 m/s for directions 1.57 rad and 2.09 rad. The different cases will have different points of reaching steady state but for these cases the steady state is reached quite early.

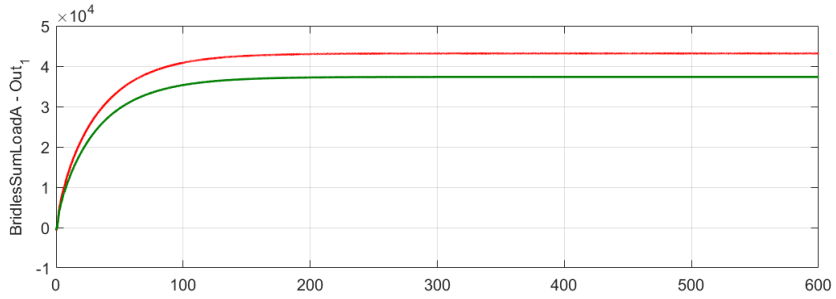


Figure 5.7: FhSim steady state 2.

5.6 Simulation Process

The simulations in this thesis were done with a regular PC and virtual machines. The virtual machines were used as it was only possible to perform 4 simulations every four hours with a PC with standard components. The simulations were sorted by naming the output files with by varying input values. The simulations and sorting in this task were done manually. This way of sorting the output files were both done to make it easier to keep track of the results but it is also useful for running the machine learning algorithm.

Name of the virtual machine	**Top Current**	**Bottom Current**	**Direction**	**Name of output file**
FhSim1-1	0.1	0.0	3.14	TopCurrent0.1 -BotCurrent0.0-Dir3.14
FhSim1-2	0.1	0.1	3.14	TopCurrent0.1 -BotCurrent0.1$_{D}ir$3.14
FhSim1-3	0.1	0.2	3.14	TopCurrent0.1 -BotCurrent0.2-Dir3.14
FhSim1-4	0.1	0.3	3.14	TopCurrent0.1 -BotCurrent0.3-Dir3.14
FhSim2-1	0.1	0.4	3.14	TopCurrent0.1 -BotCurrent0.4-Dir3.14
FhSim2-2	0.1	0.5	3.14	TopCurrent0.1 -BotCurrent0.5-Dir3.14
FhSim2-3	0.1	0.6	3.14	TopCurrent0.1 -BotCurrent0.6-Dir3.14
FhSim2-4	0.1	0.7	3.14	TopCurrent0.1 -BotCurrent0.7-Dir3.14
FhSim3-1	0.1	0.8	3.14	TopCurrent0.1 -BotCurrent0.8-Dir3.14
FhSim3-2	0.1	0.9	3.14	TopCurrent0.1 -BotCurrent0.9-Dir3.14
FhSim3-3	0.1	1.0	3.14	TopCurrent0.1 -BotCurrent1.0-Dir3.14
FhSim3-4	0.1	1.1	3.14	TopCurrent0.1 -BotCurrent1.1-Dir3.14

Table 5.3: Simulation process.

5.7 Output data

Each output data file ended up having a size of 0.25 GB, this this means that all 640 simulation cases ended up corresponded to a data size of 160 GB. These results are shown in the Appendix A. Out of all the parts contained in the output data files, in this thesis the force distribution in the bridles were chosen to use for machine learning. In the output data file the forces in the bridles are displayed in the x, y and z direction for both the connection point to the floating collar as well as the connection to the mooring. For the distribution of the bridles mooring connection forces is what has been chosen to use for machine learning. This is displayed in the output data files as ForceA.

5.7.1 Output data management

The output data files have registered data every 0,1 s for 1000 s. This results in the output data file having over 10000 rows. This file also contains unwanted data. To reduce the amount of unwanted data from the output files, a python script was used. This script removed all the other output data except Bridle ForceA for bridles 1 through 12 and the vertical displacement of the bottom of the cage. This script is shown in the Appendix B. The output data is displayed with the force distribution in the x-, y- and z-axis but it's the resultant force that is optimal to use for the machine learning algorithm. To calculate the resultant force of the bridles for each simulation output data file another python script was used. This script is also shown in the Appendix B. The formula for the resultant force for the bridles in the output data file is

$$\text{ForceA} = \sqrt{\left((\text{ ForceA }_0)^2 + (\text{ ForceA }_1)^2 + (\text{ ForceA }_2)^2\right)}.$$

When calculating the resultant force of the bridles, the average value of the force for A_0, A_1 and A_2 from 500 s to 600 s was used.

5.7.2 End result data

End result data for 24 of the 640 case simulations. These are the maximum and minimum bridle force resultant forces of all the 640 cases. All the force outputs from FhSim are given in N (Newtons). All these resultant forces were calculated using the python scripts shown in Appendix B.

	B1	B2	B3	B4
Max	52524,3	73401,7	193937,1	1379,3
Simulation	0,9_1,5_3,14	0,9_1,5_3,14	0,9_1,5_3,14	0,9_1,5_1,57

Table 5.4: Bridle forces max 1.

For Bridles 1, 2 and 3 the maximum mooring connection force is generated in the simulation case with a top current of 0.9 m/s and a bottom current of 1.5 m/s in the south direction 3.14 rad. While the maximum force generated for Bridle 4 is for the simulation case with top current of 0.9 m/s and a bottom current of 1.5 m/s in the east direction 1.57 rad.

	B5	B6	B7	B8
Max	659,1	1338,0	194391,5	73840,6
Simulation	0,0_0,1_1,57	0,9_1,5_3,14	0,9_1,5_1,57	0,9_1,5_1,57

Table 5.5: Bridle forces max 2.

For Bridle 5 the maximum force generated occurs with zero top current and a bottom current of 0.1 m/s in the east direction of 1.57 rad. For the Bridle 6 the maximum force is generated with a top current of 0.9 m/s and a bottom current of 1.5 m/s in the south direction of 3.14 rad. For Bridle 7 and 8 the maximum bridle forces are generated with a top current of 0.9 m/s and a bottom current of 1.5 m/s in the east direction of 1.57 rad.

	B9	B10	B11	B12
Max	52603,9	194493,2	73772,3	194013,2
Simulation	0,9_1,5_1,57	0,9_1,5_3,14	0,9_1,5_2,09	0,9_1,5_1,57

Table 5.6: Bridle forces max 3.

For Bridle 9 and 12 the maximum bridle force generated is with a top current of 0.9 m/s and a bottom current of 1.5 m/s in the east direction of 1.57 rad. For Bridle 11 the maximum force generated is with a top current of 0.9 m/s and a bottom current of 1.5 m/s. The same top and bottom current speeds generate the maximum force for Bridle 11 but with a direction of 2.09 rad.

	B1	B2	B3	B4
Min	88,2	188,7	384,8	50,1
Simulation	0,7_0,0_1,57	0,3_1,5_1,57	0,2_0,1_1,57	0,4_1,3_2,09

Table 5.7: Bridle forces min 1.

The minimum bridle mooring connection forces for Bridle 1, 2 and 3 is generated by a current direction of 1.57 rad. For Bridle 1 the current speed is 0.7 m/s for the top current and zero for the bottom current. For Bridle 2 the top current is 0.3 m/s and the bottom current is 1.5 m/s. For Bridle 3 the top current is 0.2 m/s and the bottom current is 0.1 m/s. For Bridle 4 the minimum force generated is with a top current of 0.4 m/s and a bottom current of 1.3 m/s with a current direction of 2.09 rad.

	B5	B6	B7	B8
Min	46,8	50,1	386,7	191,5
Simulation	0,7_0,2_2,62	0,4_1,3_2,09	0,2_0,1_3,14	0,3_1,5_3,14

Table 5.8: Bridle forces min 2.

For Bridle 5 the minimum bridle connection force occurs with a top current of 0.7 m/s and a bottom current 0.2 m/s with a current direction of 2.62 rad. For Bridle 6 the minimum force is generated with a tip current of 0.4 m/s and a bottom current of 1.3 m/s with a direction of 2.09 rad. For Bridle 7 and Bridle 8 the minimum force occurs with a current direction of 3.14 rad. For Bridle 7 the minimum force is generated with a top current of 0.2 m/s and a bottom current of 0.1 m/s. For Bridle 8 a top current of 0.3 m/s and a bottom current of 1.5 m/s.

	B9	B10	B11	B12
Min	89,1	884,4	751,0	884,5
Simulation	0,6_0,3_3,14	0,0_0,1_1,57	0,0_0,1_1,57	0,0_0,1_3,14

Table 5.9: Bridle forces min 3.

The minimum force generated for Bridle 9 is with a top current of 0.6 m/s and a bottom current of 0.3 m/s with a direction of 3.14 rad. For Bridle 10 and Bridle 11 the minimum bridle force occurs with zero top current and a bottom current of 0.1 m/s with a direction of 1.57 rad. With the same top current and bottom current the minimum bridle force for Bridle 12 is generated but with the direction of 3.14 rad.

Chapter 6

XGBoost algorithm

This chapter covers the script setup and use of the XGBoost algorithm. This includes the specifications of the algorithm as well as the result it has produced. Detailed results of predictions can be found in Appendix C. It shows the results for 20% of the total result data for Bridles 1-12 with the true force and the predicted force.

The script setup for the XGBoost algorithm can be found in Appendix B, Code listing B.3. The script imports metrics like mean absolute error (mae), and mean squared error (mse) from SciKit-Learn. This is an open machine learning library for python. The script also imports train-test-split which is a model selection from SciKit-Learn.

6.1 XGBoost setup

The mean absolute error is used to compute a risk metric which corresponds to the expected value of the absolute error loss [37]. If $\hat{y}_i$ is the predicted value of the i-th sample and the corresponding true value is y_i then the mean absolute error is defined over n_{samples} as

$$\text{mae}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| \quad [37].$$

The mean squared error is used to compute a risk metric which corresponds to the expected value of the squared error or loss [37]. The mean square error is defined as

$$\text{mse}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2 \quad [37].$$

The result data from FhSim used for the machine learning was split into 80/20. Where 80% of the result data was used for traing the machine learning algorithm and the remaining 20% was used to test the algorithms predictions. This is why the test size in the script is set to 0.2. The random state is set to 0. The random state guarantees that the splits that are generated are reproducible. The splits generated by sklearn is generated by random permutations [37]. This means that by defining the random state, the order of the various ways objects from the result data set from FhSim are selected it is possible for the random numbers to be generated in the same order every time.

The number of estimators in the script is set to 600. The estimators represents the number of trees in the machine learning model mentioned earlier in 3.6.3. The default for the XGBoost library is to set the estimators to 100 but the higher number of trees leads to higher performance of the machine learning model. Though this will lead to a a slower model [38].

To be able to control the learning rate of the machine learning model. A learning rate value is introduced. The learning rate was set to 0.05 for the XGBoost algorithm used in this thesis. This value leads to the algorithm not being able to make excessive corrections to the new trees that are added into the model [38].

Another parameter used in the XGBoost algorithm is the minimum child weights. This value represents the minimum weight needed in a child to continue the building process of partitioning. The tree partition step results inn a leaf node needs to have a higher total instance weight than the minimum child weight. With a higher minimum child weight the more conservative the algorithm will be [38]. For this machine learning model a minimum child weight of 5 has been selected.

The evaluating metrics for validation data is assigned based on the objective [38]. In this thesis regression is the objective so rmse is assigned, which is the square root of mean squared error (mse) mentioned earlier. The result data generated with FhSim was not normalized for the machine learning model in this thesis. This led to the mae an rmse becoming very large.

The depths of the trees in this machine learning model has been set as 5. The default value for XGBoost is 6 but a higher number makes the model more complex [38]. For this thesis an individual model was created for each bridle force from Bridle 1 to Bridle 12.

6.2 Algorithm results

The machine learning algorithm was able to predict the forces in the Bridles of 1-12 with an average deveation of 0.5%. These predictions were done with 20% of the random cases from the result data generated with FhSim. The true forces and prediction forces for each bridle is shown in Appendix C.

Average accuracy	101,6	Bridle 1
Average accuracy	101,2	Bridle 2
Average accuracy	100,8	Bridle 3
Average accuracy	100,6	Bridle 4
Average accuracy	99,6	Bridle 5
Average accuracy	100,2	Bridle 6
Average accuracy	100,6	Bridle 7
Average accuracy	101,7	Bridle 8
Average accuracy	101,8	Bridle 9
Average accuracy	102	Bridle 10
Average accuracy	99,4	Bridle 11
Average accuracy	96,3	Bridle 12

Table 6.1: XGBoost Results 1.

For machine learning model of the bridle forces, its Bridle 6 that has the best result with only an average deviation of 0.2%. Furthermore, the model of Bridle 3, 4, 5, 7 and 11 also have good results with only an average deviation of 0.4-0.8 %. For Bridle 1 and 2 the average deviation was 1.2% and 1.6% which was close to Bridle 8 and 9 which was 1.7% and 1.8%.

The bridle models that created the worst results were Bridle 10 and Bridle 12. Bridle 10 had an average deviation of 2%. This is a major difference compared to Bridle 6 with only 0,2%. Though the worst results was the model for Bridle 12 with an average deviation of 3.7%.

Table 6.2 previews some of the results shown in Appendix C. Here the top current, bottom current and direction is also included. The total accuracy shown in table 6.1 is the average accuracy of all the predictions for Bridle 1-12 from 20% of the random cases in the result data from FhSim.

True	Prediction	TC	BC	Dir
8280,6	7417,6	0,2	1,0	3,14
409	419,5	0,2	1,4	2,62
748,3	765,6	0,6	1,2	2,62
33822,4	32976,7	0,7	1,5	3,14
91,2	105,7	0,5	0,6	1,57
1296,9	1177,6	0,9	1,0	2,62
20435,7	20634,7	0,5	1,4	3,14
219,8	225,7	0,2	0,4	2,09
1077,2	1559,7	0,0	0,3	3,14
208,1	171,6	0,3	0,1	1,57
42686,3	43058,2	0,9	0,2	3,14
319,8	280,7	0,7	0,8	2,09

Table 6.2: XGBoost Results Bridle 1 excerpts.

These predictions are based on the previous 80% of the result data from FhSim. This means that the model has managed to predict the last 20% of cases with an accuracy of 0.5%.

Chapter 7

AKVA Group Buoy sensor data

AKVA Groups sensor buoy has registered data from early 2021 to late 2022. As an example of data input for the digital twin proposal, a small sample of the data from 01.08.21 to 31.08.21 has been used. The registered sensor buoy data is shown i figures 7.2 and 7.3. These graphs displays the different values for current direction and current velocity with five registered values each day during August 2021. By extracting a small sample of this data, it's possible to create the basis to make a framework for digital twins. More about this in chapter eight.

Figure 7.1: AKVA Group buoy in the ocean.

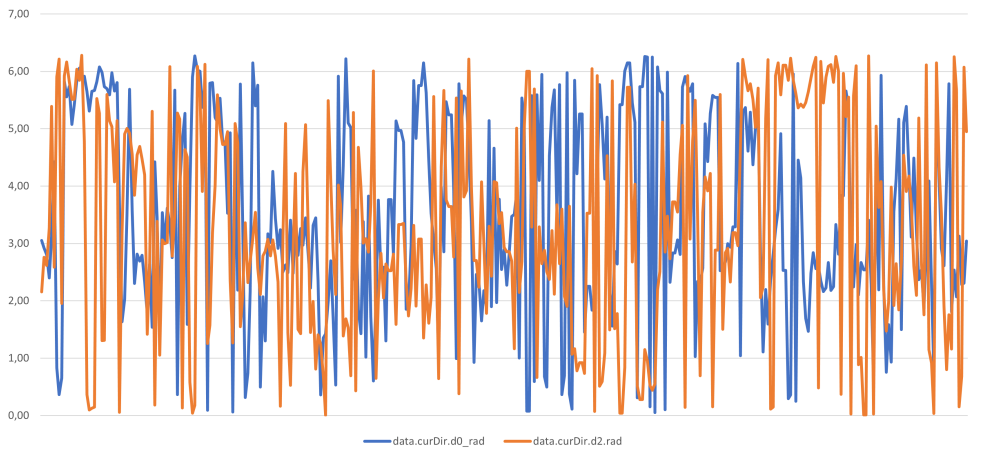


Figure 7.2: Sensor buoy current direction in radians.

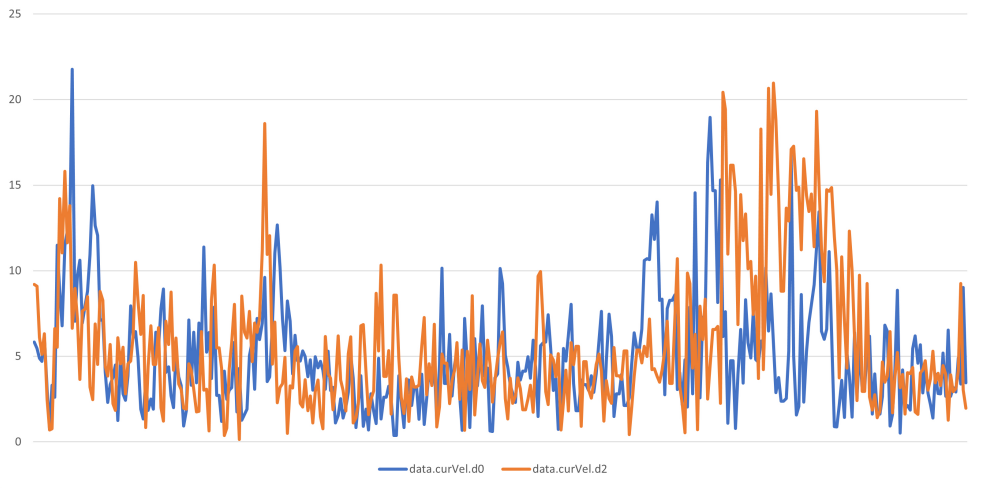


Figure 7.3: Sensor buoy current velocity in cm/s.

Chapter 8

Digital twin proposal

This chapter shows in a simple way how the machine learning model created in this thesis can be used to predict bridle forces in the net cage system by using buoy sensor data. Digital twin represents the physical reality through data and simulations. The digital twin can then result in real-time calculations, optimization and monitoring [39].

In table 8.1 selected readings from the AKVA Group sensor buoy data has been extracted. These readings were chosen because of the similarity in the current direction. At least for the d0 current direction. Considering the machine learning model is trained with only the direction moving one direction the second direction of the current d2 is not included.

Time(2021)	Dir.d0.rad	Dir.d2.rad	Vel.d0(m/s)	Vel.d2(m/s)
08-11T10:00:20Z	1,56	5,84	0,075	0,025
08-07T02:29:41Z	2,60	0,48	0,114	0,193
08-05T18:19:45Z	2,10	0,9	0,025	0,024
08-02T00:29:47Z	3,13	0,2	0,049	0,042

Table 8.1: Buoy sensor august.

On seventh of August 2021 the AKVA Buoy sensor registered a top current of 0.1 m/s and a bottom current of 0.19 m/s with the direction of 2.60 rad. These current velocities and direction are very close to the one of the cases simulated in this thesis. That is the case with top current 0.1 m/s and a bottom current of 0.2 m/s with the direction of 2.62 rad.

If these cases were to be considered the same machine learning can be used to predict the bridle forces caused by the current in the CAC net cage system at 07.08.2021.

Bridles	**True (N)**	**Prerdict (N)**	**Deviation (%)**
Bridle 1	612,6	614,8	0,4
Bridle 2	839,3	811,4	3,3
Bridle 3	4379,4	4284,8	2,2
Bridle 4	311,3	315,9	1,5
Bridle 5	260,0	260,5	0,2
Bridle 6	342,3	362,3	5,8
Bridle 7	1892,5	2203,2	16,4
Bridle 8	493,1	460,4	6,6
Bridle 9	402,0	410,1	2,0
Bridle 10	6512,8	6485,0	0,6
Bridle 11	2544,1	2680,5	5,4
Bridle 12	4232,7	4339,9	2,5
Sum			3,9

Table 8.2: ML model prediction for TC0.1, BC0.2

The machine learning model is able to predict the resultant bridle connection force to the mooring frame by an average deviation of 3.9%. This assumes that there was only current in one direction and at the same time, no waves were present that affected the net cage system.

In Table 8.2, the model shows that it can predict which bridles are exposed to the most tensile forces. For the case with top current 0.1 m/s and bottom current 0.2 m/s with the direction of 2.62 rad it's bridles 3, 7, 10, 11 and 12 which are exposed to the strongest tensile forces. For this case, it is Bridle 10 that is exposed to the most wear with a mooring connection force resultant of 6485 N.

Now if the current velocity were to change for the top current to 0.6 m/s and the bottom current to 1.2 m/s in the same direction as before the machine learning model could predict the bridle forces as follows.

Bridles	**True (N)**	**Predict (N)**	**Deviation (%)**
Bridle 1	748,3	765,6	2,3
Bridle 2	5366,6	5152,0	4
Bridle 3	96383,6	97201,7	0,8
Bridle 4	53,6	52,7	1,7
Bridle 5	47,6	52,8	10,9
Bridle 6	123,1	121,0	1,7
Bridle 7	41234,9	41617,9	0,9
Bridle 8	463,4	503,1	8,6
Bridle 9	282,7	267,1946	5,5
Bridle 10	109519,4	109367,2	0,1
Bridle 11	36142	36530,6	1,1
Bridle 12	65665,3	67160,0	2,3
Sum			3,3

Table 8.3: ML model prediction for TC0.6, BC1.2 and Dir 2.62.

The machine learning model is able to predict the resultant bridle connection force to the mooring frame by an average deviation of 3.3%. For this case the models shows that the mooring connection forces are still the strongest for bridles 3, 7, 10, 11 and 12. Though now the mooring connection forces have been increased for these bridles but shows the remaining bridles with weaker mooring connection forces.

By using machine learning models such prediction can be made for several current velocities at different depths in different directions. A digital twin can then be made for the net cage system with the machine learning model developed in this thesis. Though the machine learning model developed is only able to predict accurate results for cases that is simulated within the current range and direction performed in this thesis.

Input data from the sensor buoy can then be transferred to the machine learning model before the output data is transferred to the digital twin and stored for complete overview. With long-term stored data and history of the bridle forces in the net cage system, worn or damaged parts can be repaired or replaced. If currents at different depths with different velocities in different directions other than the range of simulations in this thesis are to be considered, additional simulations must be performed.

Chapter 9

Discussion

This chapter discusses and evaluates the content of this thesis. Among other things, various shortcomings and changes that could have been made to improve the results are discussed.

The FhSim simulation results is discussed by looking at the different bridles and their maximum and minimum mooring connection forces and what simulation case these maximum values are connected to. The discussion also covers how these maximum and minimum values for each bridle can be used.

Machine learning results are also discussed. This section includes the accuracy of the machine learning model for two of the different cases presented in the chapter 8. An evaluation of the parameters used in the XGBoost script are also discussed. A section with the exclusion of waves are also discussed in this chapter. This discussion includes how the waves would affect the net cage system with different wave directions if they were to be simulated with the current.

Further the quantity of FhSim simulations are discussed. This section includes solution of how symmetry and mirroring can be used for the net cage system to reduce the amount of simulations which needs to be implemented.

The chapter also includes a section with a discussion about reducing faulty data registered by the buoy sensor. Lastly the chapter covers discussion on the use of machine learning for aquaculture net cage systems and the quality of the simulation output generated with FhSim in this thesis.

9.1 FhSim simulation results

In the 24 results shown in 5.7.2 the bridle connection force to the mooring frame is largest for Bridle 3, Bridle 7, Bridle 10 and Bridle 12. These bridles experiences a connection force of almost 200 kN each. By comparing the maximum and minimum values, a clear overview is formed of how large differences the current speed and direction can have for the connection forces.

With a top current speed of 0.9 m/s and a bottom current speed of 1.5 m/s with a direction of 3.14 rad, Bridle 3 experiences a connection force of 194 391.5 N. While with a top current speed of 0.2 m/s and a bottom current speed of 0.1 m/s with a direction of 1.57 rad, Bridle 3 only experiences a connection force of 384,8 N. That's a difference of almost 194 kN. The same applies to Bridle 10. Where the maximum connection force is also almost 194 kN larger than the minimum connection force value.

The difference in the magnitudes of the maximum value of the connection force is also interesting. It is quite clear that the connection forces for Bridles 4, 5 and 6 are a lot smaller than the maximum connection forces for the other bridles. This is due to the chosen directions for the simulations. Bridles 4, 5 and 6 are the bridles that will experience the smallest tensile forces. This is because the net cage system drifts with the current in the direction that Bridles 4, 5 and 6 are located.

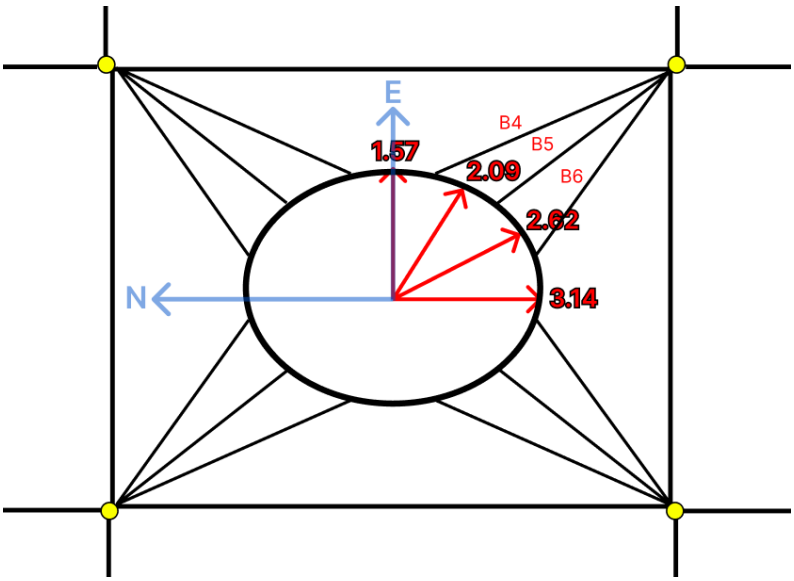


Figure 9.1: FhSim directions with B4, B5 and B6.

The results in FhSim makes it quite clear that there are large variations for the force distribution that each bridle is exposed to in the net cage system. If one were to estimate this force distribution it would be very difficult to do this based on experience and assumptions. Therefore, a setup where numerical simulation is used can be a good solution. At the same time, it can be very expensive to get professionals to design the net cage system in FhSim. But by experimenting with the net cage system with numerical simulation, savings can be made for the design of the net cage system.

As for Bridle 3, 7, 10 and 12 where it is shown that with high current speed in a certain direction the force distribution in these bridles will increases considerably. Then it is possible to design these bridles to be more suitable for stronger forces and other bridles such as Bridle 4, 5 and 6 can be designed to be able to handle minor, less strong forces.

If the location that the net cage system is going to be placed has a known specific average current velocity and direction, the net cage system can be designed accordingly. For example a location with a current direction averaging from 1,57 rad to 3,14 rad. With a current velocity ranging from 0.0 m/s to 0.9 m/s for the top current and 0.0 m/s to 1.5 m/s for the bottom current. Bridles 3, 7, 10 and 12 would need to be made stronger. At least stronger than bridles 1, 2, 8 and 9 which again would need to be made stronger than bridles 4, 5 and 6.

9.2 Machine learning results

The machine learning result clearly shows that it is possible to make predictions for bridle forces based on data from previous simulations. At the same time, there are still deviations in the model that need to be improved.

For the prediction of the case with top current 0.1 m/s and a bottom current of 0.2 m/s with a direction of 2.62 rad, Bridle 7 is the bridle with the largest discrepancy in machine learning prediction. This may be due to the wide variety of force Bridle 7 experiences shown in chapter 5.7.2. On the other hand, Bridle 7 is also exposed to the same variety of force as Bridle 10, but Bridle 10 still has a very good results in the machine learning model with only a deviation of 0.6%.

Another reason why Bridle 7 may have such a large discrepancy is that the 80/20 distribution has not distributed all the cases evenly in the machine learning model. This may mean that the machine learning model has received a large selection of training data for Bridle 10 as Bridle 10 has a very good prediction, but much of the data that affects the prediction results of Bridle 7 has been used in test data. So the model is not optimized to make accurate predictions for Bridle 7 for this specific case.

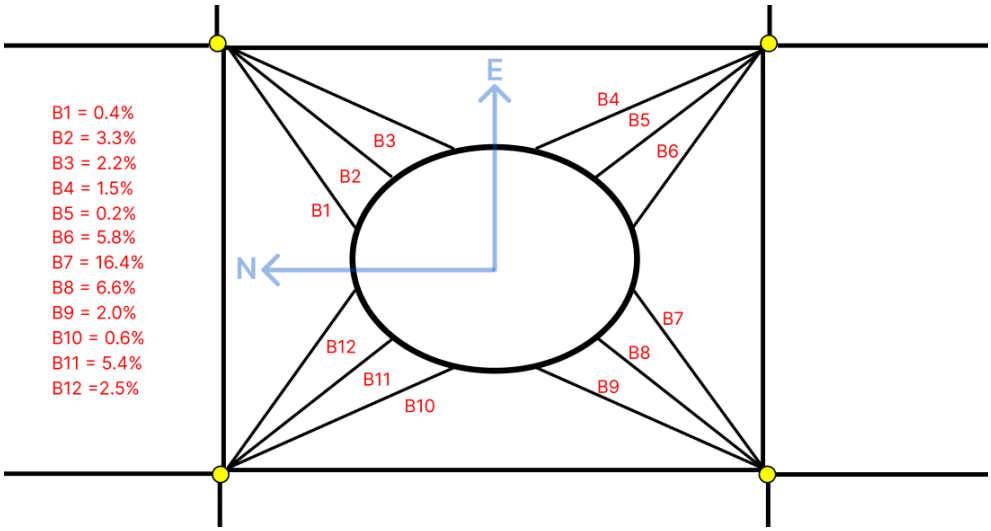


Figure 9.2: XGBoost deviation 1.

For the prediction case with top current 0.6 m/s and bottom current 1.2 m/s with a direction of 2.62 rad, Bridle 5 is has the largest discrepancy in machine learning prediction. For this case, it is Bridle 5 that had the largest deviation of 10.9 %. Meanwhile for this case Bridle 7 is at 0.9 %.

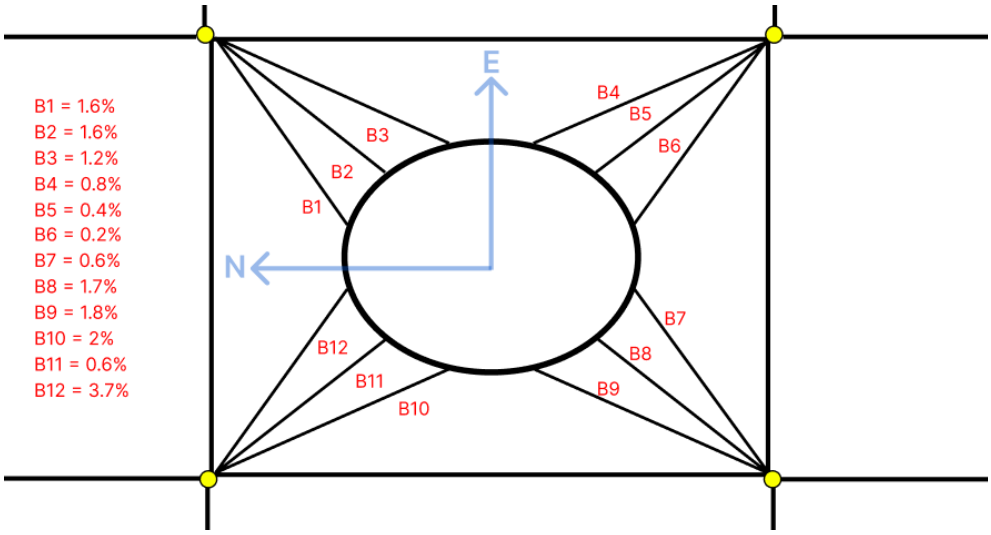


Figure 9.3: XGBoost deviation 2.

It is quite clear that if a digital twin is to be formed that the machine learning model must be trained with a larger data sets. As the model is now, there are only 128 cases per bridle. This makes it difficult to combine with sensor buoy data as the data from the buoy has millions of different combinations. It is also possible to make assumptions that approximately the same current direction and speed will lead to the same loads. As done in Chapter 8.

It could also be possible to optimize the machine learning model by setting up the number of trees from 600 to 700 and changing the depth of these trees from 5 to 6. At the same time, it is not certain that this would lead to a more accurate model. For the result shown in Tables 8.2 and 8.3, current moves in the same direction with only a change in current speed. Nevertheless, there is a big difference in how accurate the machine learning model is for the different bridles. In Table 8.2, Bridle 7 has the largest deviation, but for Table 8.3, the prediction made by the model is much more accurate. There is thus a prediction accuracy difference of 15.5 % for Bridle 7 between the two cases.

The training data in the model remains the same and with such large differences for the bridle prediction for each case, extended tweaking of the machine learning model would not be the most optimal way of increasing the machine learning models accuracy. Thus in order to improve the machine learning model, a better way of increasing the accuracy would be to perform more simulations and gather data that can be used for training and testing of the machine learning model.

9.3 Excluding waves

Although it's quite unrealistic to envisage a net cage system without waves, it does not necessarily lead to the machine learning result becoming unrealistic. The purpose of this thesis is to show the correlation between simulation data and the result from output data used with machine learning model. It's quite clear that if a very accurate machine learning algorithm were being developed, a more detailed simulation model would be absolutely necessary. An important part of the net cage system is the floating collar and it is the wave force that exposes the collar to most wear. At the same time, it is not certain that waves with constant significant wave height and wave length would lead to more accurate results for the machine learning algorithm. It is the variation of parameters that makes the difference for a more detailed simulation model.

If waves were to be included in the simulations, the bridle forces could either have been increased or decreased in the simulation result. This is because the waves could have been simulated either against or with the current, it could also have been simulated going across the current. Waves against current could lead to less tensile force in bridles that are on the same side as the current direction. In figure 9.4 the bridle forces for B7-B12 are the highest, but with a wave force that moves the floating collar against the current the bridles forces in B7-B12 could be reduced. Though this would lead to bigger displacement of the net cage.

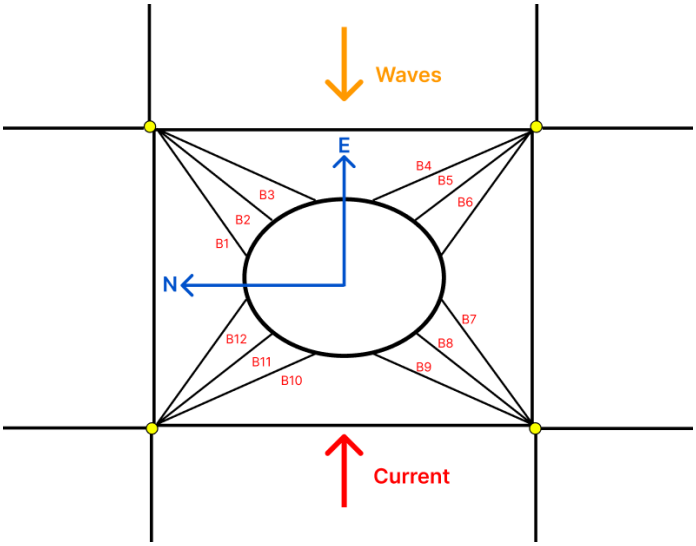


Figure 9.4: FhSim Wave direction 1.

If the simulations included waves that moved in the same direction as the current, it would have led to increased bridle force in some bridles. In figure 9.5 the increased tensile force would be in bridles B7-B12.

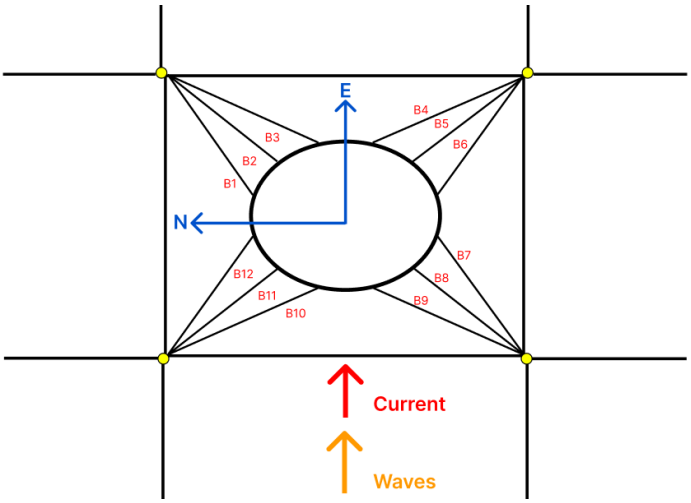


Figure 9.5: FhSim Wave direction 2.

If the simulation included waves that were crossing the current direction, this would also lead to a change in the tensile force for the already exposed bridles. In figure 9.6 this would be bridles B10-B12 and B1-B3.

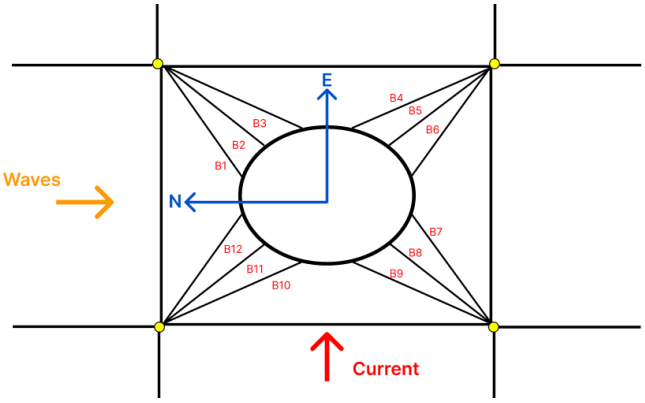


Figure 9.6: FhSim Wave direction 3.

Waves can therefore make a big difference to the simulation result and should be included in future more accurate simulations, but for this task it was not necessary. That is because with waves, the model would be more difficult to use together with symmetry to show that it's possible to mirror the bridle force distribution for the current directions.

9.4 FhSim simulation quantity

One could argue that 640 simulations is not a sufficient number of simulation output data for machine learning. This is because a larger amount of training data would lead to a more accurate machine learning algorithm. The problem, however, is that the number of simulations increases drastically with more parameters that are introduced in the model. This applies to the number of directions, waves and wave setup, current and current direction. In this thesis there were 10 current velocities for the top current and 16 current velocities for the bottom current. These were split up into four different directions. This resulted in 640 simulations. If one were to introduce a current velocity from 0 m/s to 0.9 m/s with 0.01 m/s increments rather than 0.1 m/s increments for the top and bottom current. The total amount of simulations would be around 55000. If the current directions also would increase from four different directions to 12 directions. The number of simulations would be around 165 000. In order to have a very detailed machine learning model, this will therefore require a larger time frame and would need collaboration with others to a higher degree than what has been available for this master's thesis.

To make the machine learning model more accurate than what has been done in this thesis, the current directions could be distributed in a better way. This is because in this thesis two of the directions that are simulated give the result of the other two directions via symmetry. Although this shows that the simulations have been carried out correctly, this does not create greater variation to aid the machine learning algorithm.

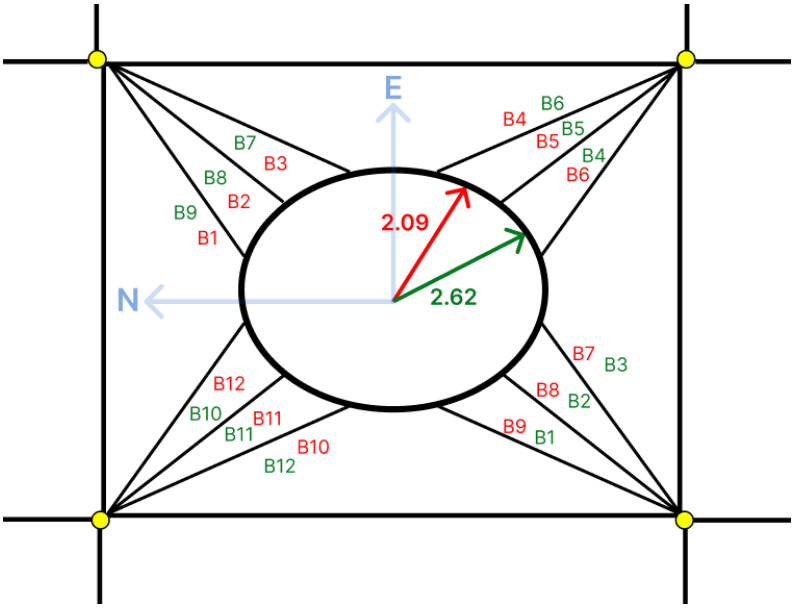


Figure 9.7: FhSim bridle symmetry mirroring.

Figure 9.7 shows how the symmetry of bridles will change if the current direction 2.09 rad is set to 2.62 rad. Although this is could be negative for the variation in the machine learning algorithm, it was easier to carry out the simulations in this manner than to process it afterwards.

Table 9.2 and 9.3 proves the ability to mirror the net cage system by using symmetry. As seen in figure 9.5 the bridles B11 and B5 will have the same force value when using symmetry. This can also be shown from the result files in Appendix A. For a simulation case with top current 0.2 m/s and bottom current 0.5 m/s and with a current direction 2.09 rad and 2.62 rad. The forces in bridle 5 and 11 are the same. As long as the top and bottom currents are the same for current directions 2,09 rad and 2,62 rad the bridles can be mirrored.

Case/Bridle	B1	B2	B4	B5	B11	B12
0,2_0,5_2,09	208,3	278,0	206,6	148,6	7378,8	24966,1
0,2_0,5_2,62	447,1	844,2	180,0	148,5	7378,8	15066,9
0,8_0,8_2,09	404,9	670,1	163,2	76,8	50809,1	143857,0
0,8_0,8_2,62	1054,1	12184,6	53,5	76,4	50845,8	87575,6

Table 9.1: Bridle symmetry 1.

Case/Bridle	B1	B2	B4	B5	B11	B12
0,2_0,5_2,09	208,3	278,0	206,6	148,6	7378,8	24966,1
Case/Bridle	B9	B8	B6	B5	B11	B10
0,2_0,5_2,62	208,6	278,4	206,3	148,5	7378,8	24931,7

Table 9.2: Bridle symmetry 2.

Case/Bridle	B1	B2	B4	B5	B11	B12
0,8_0,8_2,09	404,9	670,1	163,2	76,8	50809,1	143857,0
Case/Bridle	B9	B8	B6	B5	B11	B10
0,8_0,8_2,62	406,1	671,1	162,3	76,4	50845,8	143856,0

Table 9.3: Bridle symmetry 3.

The values are slightly different, but are similar enough for the difference to be negligible. A better choice would therefore be to divide the current directions into half of a circle quadrant. This way the force distribution in the bridles created by these current directions could have been mirrored for the whole net cage system by using symmetry. This presupposes that only the current exposes the net cage system to environmental forces.

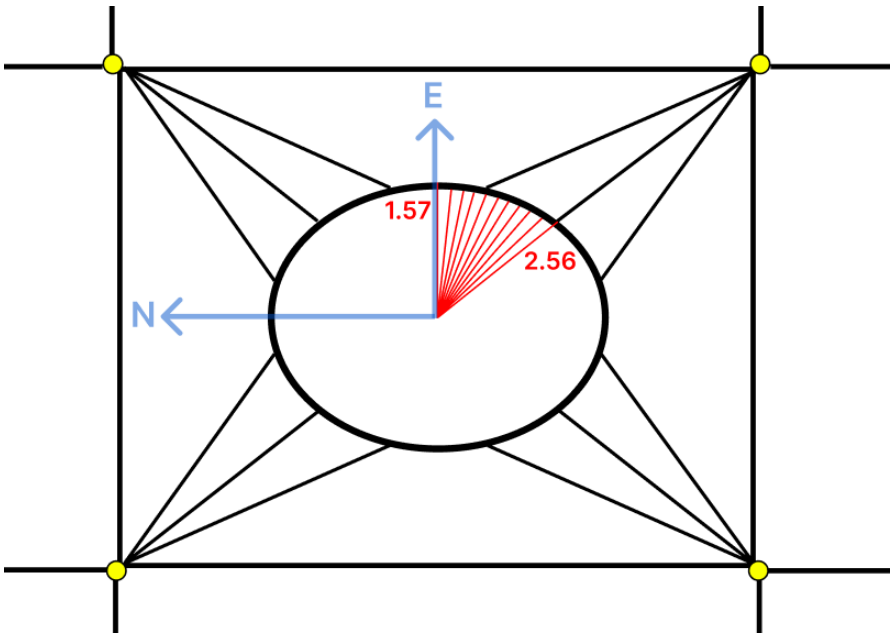


Figure 9.8: FhSim half circle quadrant.

9.5 Noise reduction for sensor data

When it comes to recording sensor data, the sensors registered data is not necessarily error-free. The data may have discrepancies. These incorrect data registrations can be referred to as sensor noise. Almost all sensors can experience some type of sensor noise in its data.

For AKVA Group's oceanographic sensor buoy, there may be several different factors that can lead to sensor noise. This may be because of problems with the battery power level or because of fish or other living creatures in the ocean that can interfere with the sensors. Sensor noise can also be caused by the transmission of information between the sensor buoy and the reception center. For example, in this thesis, errors occurred in csv output files that were compressed and uploaded to cloud storage. When these csv files were downloaded from the cloud storage, the data in some of the csv files were incomplete.

Another factor that can lead to sensor noise are boats and other maritime traffic. To create an accurate digital twin framework, precise sensor data is needed. It is thus important to take into account that there will be sensor noise in registered data from AKVA Group's oceanographic buoy and measures must be taken to deal with this sensor noise.

To reduce sensor noise for the oceanographic buoy, it may be appropriate to consider the location of the buoy. It is also possible to use various statistical and probabilistic methods to reduce sensor noise. These statistical and probabilistic methods will then be able to distinguish abnormal readings registered by the sensor buoy. Though it's still not certain that the statistical methods may find all the divergences in the registered data. It may therefore be most beneficial to consider location as well as other environmental influences that may cause sensor noise as well.

In this thesis, the time period for registered data by AKVA Group's sensor buoy is not very long. It is therefore easier to detect abnormal readings manually, rather than using statistical and probabilistic methods. If the loads on the components in the CAC net cage system are to be mapped based on the data that AKVA Group's sensor buoy has gathered since it was deployed, it will be beneficial to use statistical and probabilistic methods to adjust for the sensor noise.

9.6 Machine learning for aquaculture net cage systems

In collaboration with AKVA Group, the idea of digital twins with correct information about force distribution for the net cage system has been very well received. One could argue that this type of technology will be more costly than it's degree of usefulness, but the technology can also be used for development of the net cage system. If the digital twin is able to develop a maintenance pattern, it can also be used to get an insight into what components of the net cage system needs to be strengthened.

This in turn can lead to better design of net cage systems which can possibly be used in areas where the environmental forces are similar. Although it can be expensive to develop such technology, it can lead to reduced maintenance on net cage systems which in turn leads to cost savings in the use phase as well as during the design phase.

For fish farming, the risk of fish escaping is of great interest. With today's maintenance solution, abnormal weather conditions can lead to greater wear on the net cage system, which can be overlooked by professionals who assumes the wear of the components based on previous experience. These assumptions based on previous experience could prove insufficient for an accurate maintenance model. With machine learning technology, the process would be automatic and more accurate. This reduces the degree of expertise required to understand the maintenance or replacement of components for the net cage system.

With known force distribution in the net cage system, the process of choosing the appropriate optimal location for the net cage system could also be simplified. This can lead to an overall improvement of suitable components and optimal location for the net cage system which can lead to an overall cost reduction.

As mentioned in 9.2, the best measure to make the machine learning model, is to collect more result data for training and testing the model. One way to do this is to apply the symmetry principle shown in 9.4. The principle in 9.4 could then add waves and different wave specifications, different current velocities for different depths. Then use symmetry to mirror the mooring connection bridle forces onto the rest of the net cage system.

9.7 Quality of FhSim output results

The simulations in this thesis were run manually. That means the input file document for FhSim was modified 640 times, the same applies to the storage of output documents. In the output file from FhSim, the input values are not specified. This means that the validity of the output data is based entirely on the manual process of naming the file correctly. With such a high number of simulations this leads to a set of certain uncertainties.

This leads to uncertainty about the outcome of the machine learning model. At the same time, the machine learning model showed successful results for several of the cases in FhSim. Though it may be that deviations in the machine learning model could be caused by errors in the manual simulation and storage of data. Since predictions for several of the cases do not constitute a deviation that diverges greatly in the result, this will most likely mean that no major errors have been made in the manual process around the simulation data.

Another uncertainty for the simulation result is the resultant calculation done with python script. It may be that there is incorrectly placed data in the raw data result from FhSim. When using python script to process csv files, these files are not opened. It may therefore be that errors in the files have not been observed.

A way to further improve the development of the machine learning model could therefore be to control the simulation results from FhSim more accurately.

Chapter 10

Conclusion

In this thesis, numerical simulation, sensor data and machine learning have been used to investigate the possibility of creating a digital twin framework for net cage systems. The result of the thesis clearly shows that this is possible. This is new research that has not been proven before. With an average deviation of 0.5%, further development of the machine learning model can lead to a very accurate digital twin. In the machine learning model, there are several cases where predictions are more inaccurate than others, but the total deviation shows an acceptable result. This technology can thus be used to make aquaculture in Norway more efficient. Instead of basing maintenance on estimates and previous experience, wear on components in the net cage system can now be logged and displayed in a digital twin.

As mentioned in Chapter 1.1, the annual growth rate of fish consumption has outpaced the annual population growth rate from 1961-2017. With a world population that continues to increase, it will be absolutely necessary to optimize the fish farming industry in the world. In Norway, this can be done by maximizing the utilization of the structure of traditional gravity net cage systems. This is done by making maximum use of the components in the the net cage system in the design phase, the use phase and the maintenance phase during the net cage system's lifetime.

In the design phase of the net cage system numerical simulation can be used to give an estimate of how the location of the net cage system will affect the components. During the use phase numerical simulation combined with sensor data and machine learning can be used to predict the distribution of forces in the net cage system. When the components then need to be maintained, the machine learning model which can be used for the digital twin knows what components has been exposed to the greatest force over a certain time period. When these components are to be replaced, the machine learning model and digital twin can also be used to understand whether the components may need to be replaced with a stronger material or not.

Chapter 11

Further work

This chapter proposes further work based on findings made in this thesis. Python scripts and simulation results are attached in Appendices A, B and C, so that future similar research can be completed without having to repeat these simulations. This thesis has a wide range, and has led to many ideas for further work which have emerged along the way.

This includes looking at different current velocities for different depths, investigating the possibility of changing the starting position in the simulation of the net cage system where the last one ended, reversing the machine learning model, optimizing the simulation software for machine learning, improving the digital twin by using other types of sensors and developing a more accurate machine learning model by performing simulations with slower current velocities.

11.1 Advanced currents for net cage systems

In this thesis, investigations were only made for different current velocities with the same current direction at different depths. For future work, there may be a possibility to change the FhSim setup to do simulations with different current depths and velocities.

The AKVA Group sensor buoy registers data at three different depths $5m$, $10m$ and $15m$. In order to obtain a more accurate result, it could therefore be interesting to investigate how the result changes with three different current depths instead of two. This in combination with the current velocity being different for all three directions. This will result in a larger number of simulations as well as simulation time, but it can produce a very thought-provoking result. It may be more beneficial to simplify the investigations by first examining different current velocities for three different depths with the same current direction. It will also be wise for future work to base the simulations on current velocity sensor data that already exists.

It may also be interesting to investigate simulations where the current changes during a certain period of time. Then a result that is more realistic can be used in a digital twin. These changes can then be based on average current changes that are registered by the sensor buoy. Studies where current at different depths decreases or disappears after a certain time period can also be good to include in a machine learning model which can be used for the development of digital twins.

11.2 Accurate starting position

Another exciting topic that can be investigated in further work is the possibility that the simulations can start where the previous one left off. This means that if the current velocity or direction changes in the net cage system, the next case simulation can start from where the previous simulation ended.

When the simulations are restarted from the zero position, loads on bridles and other components in the net cage system may not be included. It is also possible that by always starting the simulations from the zero position, that some bridles experience less force distribution than what they are actually exposed to.

11.3 Reversing the machine learning model

For this thesis, a study on whether it is possible to predict force distribution on bridles in the net cage system by making simulations based on input values in the form of current velocity and direction have been performed. A suggestion for further work may be to reverse the way of conducting machine learning on such net cage systems.

This can be done by examining the relationship between bridle forces instead of examining the relationship between environmental parameters and the force distribution in the bridles.

To get accurate readings of force distribution in bridles, shackle sensors can be used. Then a series of readings of these shackle sensors can be performed. When enough data has been registered about the forces in all the 12 bridles, machine learning can be used. Then it may be possible to use only two load shackle sensors to predict the rest of the force distribution in the net cage system for the other bridles. Experiments with this technique can also be performed using other machine learning algorithms than XGBoost.

11.4 Optimizing FhSim for machine learning

The results have shown that it is possible to use machine learning models together with FhSim. In order for better machine learning models to be developed, more data will be required. Therefore, it may be relevant for future work to create a layout that makes it possible to change input parameters faster. This can be done with python scripts or other methods.

The same applies for the sorting of the output data. All the 640 simulations in this thesis were moved, named and sorted manually. This was very time consuming. Since the output files are relatively large, i.e about 0.25 GB each, transferring these output files was also very time consuming. This is therefore also something that can be considered in further work.

11.5 Reduced current velocity

If the machine learning model is to be used for more registrations made by AKVA Group's sensor buoy, more data from simulations with slower current velocities will be needed. In this thesis, the simulations were based on minimum and maximum values of the AKVA Group's sensor buoy. This led to execution of simulations with current velocity intervals that were at a higher level than most of the current velocities that was registered around the CAC project.

Bibliography

[1] FAO, *The state of world fisheries and aquaculture 2020. sustainability in action, rome.* (Accessed: 05.09.2021), 2020. [Online]. Available: https://doi.org/10.4060/ca9229en.

[2] FAO, *The state of world fisheries and aquaculture 2018 – meeting the sustainable development goals. rome*, (Accessed: 05.09.2021), 2018a. [Online]. Available: www.fao.org/3/i9540en/i9540en.pdf.

[3] FAO, *The state of world fisheries and aquaculture*, (Accessed: 07.09.2021), 2010. [Online]. Available: https://www.fao.org/3/i1820e/i1820e.pdf.

[4] N. og fiskeridepartementet, *Forskrift om krav til teknisk standard for flytende akvakulturanlegg (nytek-forskriften)*, (Accessed: 13.05.2022), 2012. [Online]. Available: https://lovdata.no/dokument/SF/forskrift/2011-08-16-849#KAPITTEL_8.

[5] Norsk Standard., *Floating aquaculture farms — site survey, design, execution and use*, (Accessed: 13.05.2022), 2021. [Online]. Available: https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=1367329.

[6] O. Lekang, *Aquaculture Engineering, Second edition, p. 249-285*. John Wiley Sons, 2013.

[7] Osnes, A., *Groe*, (Accessed: 23.09.2021), 2021. [Online]. Available: https://snl.no/groe_-_p%5C%C3%5C%A5_skip.

[8] AKVA Group, *Polarcirkel plastic pens*, (Accessed: 24.09.2021), 2021. [Online]. Available: https://f.hubspotusercontent30.net/hubfs/4074933/Produktark/Polarcirkel%5C%20500R%5C%20Combi_EN.pdf.

[9] Aquanet, *Kobberbasert notbeskyttelse*, (Accessed: 08.02.2022), 2022. [Online]. Available: https://www.aquanet.no/kobberbasert.html.

[10] Polyform, *Buoy*, (Accessed: 08.02.2022), 2022. [Online]. Available: https://polyform.no/hard-shell-pe-products/aqua-series/.

[11] Hao Tang, Liuxiong Xu and Fuxiang Hu, *Hydrodynamic characteristics of knotted and knotless purse seine netting panels as determined in a flume tank*, (Accessed: 21.05.2022), 2018. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0192206.

[12] 0. M. Faltinsen, *SEA LOADS ON SHIPS AND OFFSHORE STRUCTURES, p. 174-175*. Cambridge University Press 1990, 1990.

[13] Pascal Klebert, Pal Lader, Lars Gansel and Frode Oppedal, *Hydrodynamic interactions on net panel and aquaculture fish cages: A review*, (Accessed: 21.05.2022), 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801812003940.

[14] Hui Chenga„ Lin Lia, Karl Gunnar Aarsætherb, Muk Chen Ongl, *Typical hydrodynamic models for aquaculture nets: A comparative study under pure current conditions*, (Accessed: 21.05.2022), 2020. [Online]. Available: https://www.researchgate.net/publication/339981742_Typical_hydrodynamic_models_for_aquaculture_nets_A_comparative_study_under_pure_current_conditions.

[15] Hui Chenga„ Lin Li, Karl Gunnar Aarsætherb, *Numerical investigation on the cage-to-cage wake effect: A case study of a 4 × 2 cage array*, (Accessed: 22.05.2022), 2021. [Online]. Available: https://www.researchgate.net/publication/348611539_Numerical_Investigation_on_the_Cage-to-Cage_Wake_Effect_A_Case_Study_of_a_4_2_Cage_Array.

[16] Odd M. Faltinsen, Yugao Shen, *Wave and current effects on floating fish farms*, (Accessed: 22.05.2022), 2018. [Online]. Available: https://link.springer.com/article/10.1007/s11804-018-0033-5.

[17] Sintef Ocean, *Fhsim*, (Accessed: 27.09.2021), 2021. [Online]. Available: https://www.sintef.no/programvare/fhsim/.

[18] Reite, K.-J., Føre M., Aarsæther K. G., Jensen J., Rundtop P., Kyllingstad L. T., Endresen P. C., Kristiansen D., Johansen V. Fredheim A., *Fhsim — time domain simulation of marine systems*, (Accessed: 14.10.2021), 2014. [Online]. Available: https://www.researchgate.net/publication/285209093_FHSIM_-_Time_Domain_Simulation_of_Marine_Systems.

[19] Svibs, *Operational modal analysis*, (Accessed: 16.10.2021), 2021. [Online]. Available: http://www.svibs.com/resources/ARTeMIS_Modal_Help/webframe.html#Operational%5C%20Modal%5C%20Analysis.html.

[20] Priour D., *Introduction of mesh resistance to opening in a triangular element for calculation of nets by the finite element method*, (Accessed: 17.10.2021), 2001. [Online]. Available: https://www.researchgate.net/publication/245450673_Introduction_of_mesh_resistance_to_opening_in_a_triangular_element_for_calculation_of_nets_by_the_finite_element_method.

[21] Faltinsen O. M. Kristiansen T., *Experimental and numerical study of an aquaculture net cage with floater in waves and current*, (Accessed: 18.10.2021), 2014. [Online]. Available: https://www.researchgate.net/publication/269820995_Experimental_and_numerical_study_of_an_aquaculture_net_cage_with_floater_in_waves_and_current.

[22] ScaleAQ, *Sinker tube*, (Accessed: 20.10.2021), 2021. [Online]. Available: https://scaleaq.com/product/sinker-tube/?cn-reloaded=1.

[23] Endresen P. C., Birkevold J., Føre M., Fredheim A., Kristiansen D. Lader P., *Simulation and validation of a numerical model of a full aquaculture net-cage system*, (Accessed: 20.11.2021), 2014. [Online]. Available: https://asmedigitalcollection.asme.org/OMAE/proceedings-abstract/OMAE2014/45493/V007T05A006/278544.

[24] Tidemann A. Elster A. C., *Maskinlæring*, (Accessed: 05.11.2021), 2019. [Online]. Available: https://snl.no/maskinl%5C%C3%5C%A6ring.

[25] MathWorks, *Machine learning in matlab*, (Accessed: 06.12.2021), 2021. [Online]. Available: https://ch.mathworks.com/help/stats/machine-learning-in-matlab.html.

[26] Seldon, *Machine learning regression explained*, (Accessed: 26.05.2022), 2019. [Online]. Available: https://www.seldon.io/machine-learning-regression-explained.

[27] ZIZMALL, *Application of linear regression in machine learning*, (Accessed: 26.05.2022), 2022. [Online]. Available: https://www.zizmall.com/?category_id=4123163.

[28] Jason Brownlee, *A gentle introduction to the gradient boosting algorithm for machine learning*, (Accessed: 26.05.2022), 2016. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[29] Christophe Pere, *A gentle introduction to the gradient boosting algorithm for machine learning*, (Accessed: 26.05.2022), 2020. [Online]. Available: https://towardsdatascience.com/what-is-loss-function-1e2605aeb904.

[30] Ivanna Baturynska Kristian Martinsen, *Prediction of geometry deviations in additive manufactured parts: Comparison of linear regression with machine learning algorithms*, (Accessed: 26.05.2022), 2021. [Online]. Available: https://www.researchgate.net/publication/340524896_Prediction_of_geometry_deviations_in_additive_manufactured_parts_comparison_of_linear_regression_with_machine_learning_algorithms.

[31] Amazon web services, *How xgboost works*, (Accessed: 26.05.2022), 2022. [Online]. Available: https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html.

[32] AKVA Group, *Akva oceanographic environmental buoy*, (Accessed: 28.02.2022), 2021. [Online]. Available: https://www.akvagroup.com/sea-based-aquaculture/camera-sensors/enviro-sensors-/akva-oceanographic-environmental-buoy.

[33] AKVA Group, *Akva group safe guard*, (Accessed: 20.11.2021), 2020. [Online]. Available: https://www.akvagroup.com/Downloads/Datasheets/AKVA%5C%20Safe%5C%20Guard%5C%20NO.pdf.

[34] ilaks, *Å få til dette i sjø under naturlige produksjonsbetingelser, der vi har vær og vind og sykdom og lus, er en milepæl*, (Accessed: 28.02.2022), 2020. [Online]. Available: https://ilaks.no/a-fa-til-dette-i-sjo-under-naturlige-produksjonsbetingelser-der-vi-har-vaer-og-vind-og-sykdom-og-lus-er-en-milepael/.

[35] Egersund, *Coned nets*, (Accessed: 20.05.2022), 2022. [Online]. Available: https://www.egersundnet.no/products/fish-farming-nets/coned-nets.

[36] AS Fiskevegn, *Danline*, (Accessed: 26.05.2022), 2022. [Online]. Available: https://www.fiskevegn.no/product/danline/.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, 'Scikit-learn: Machine learning in Python,' *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[38] dmcl, *Xgboost parameters*, (Accessed: 23.05.2022), 2020. [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html.

[39] Rasheed A., San O. Kvamsdal T., *Digital twin: Values, challenges and enablers from a modelling perspective*, (Accessed: 23.10.2021), 2020. [Online]. Available: https://www.sintef.no/publikasjoner/publikasjon/1783639/.

Appendix A

Simulation result

Underneath are two hyperlinks that is used as a forward to the data sheets.

Simulation results, raw data

Resultant forces for Bridle 1 to Bridle 12.

Appendix B

Python Scripts

Code listing B.1: Python script for csv cleaning

```
import os
import pandas as pd


files = os.listdir('./data')

header_keep = [
    'NetStructure_BottomPos_2',
    'Bridle1_ForceA_0',
    'Bridle1_ForceA_1',
    'Bridle1_ForceA_2',
    'Bridle2_ForceA_0',
    'Bridle2_ForceA_1',
    'Bridle2_ForceA_2',
    'Bridle3_ForceA_0',
    'Bridle3_ForceA_1',
    'Bridle3_ForceA_2',
    'Bridle4_ForceA_0',
    'Bridle4_ForceA_1',
    'Bridle4_ForceA_2',
    'Bridle5_ForceA_0',
    'Bridle5_ForceA_1',
    'Bridle5_ForceA_2',
    'Bridle6_ForceA_0',
    'Bridle6_ForceA_1',
    'Bridle6_ForceA_2',
    'Bridle7_ForceA_0',
    'Bridle7_ForceA_1',
    'Bridle7_ForceA_2',
    'Bridle8_ForceA_0',
    'Bridle8_ForceA_1',
    'Bridle8_ForceA_2',
    'Bridle9_ForceA_0',
    'Bridle9_ForceA_1',
    'Bridle9_ForceA_2',
    'Bridle10_ForceA_0',
    'Bridle10_ForceA_1',
    'Bridle10_ForceA_2',
    'Bridle11_ForceA_0',
```

```
    'Bridle11_ForceA_1',
    'Bridle11_ForceA_2',
    'Bridle12_ForceA_0',
    'Bridle12_ForceA_1',
    'Bridle12_ForceA_2',
    'BridlesSumLoadA_Out_0',
    'BridlesSumLoadA_Out_1',
    'BridlesSumLoadA_Out_2',
            ]

for file in files:
    data_path = './data/' + file
    df = pd.read_csv(data_path, header=None, sep=';')

    # there are two rows of header in raw data
    row_one = df.iloc[0].values.tolist()
    row_two = df.iloc[1].values.tolist()

    # re-organize the two-row header to one-row
    header = []
    for i in range(len(row_one)):
        a = row_one[i]
        b = str(row_two[i])
        if b == 'nan':
            header.append(row_one[i])
        else:
            header.append(row_one[i] + '_' + row_two[i])

    # remove the old header
    df.drop(df.head(2).index, inplace=True)
    # change the header
    df.columns = header
    # select the features we need
    df_select = df[header_keep]
    # save new csv file

    new_data_folder = './new_data/'

    if not os.path.exists(new_data_folder):
        os.makedirs(new_data_folder)

    df_select.to_csv(new_data_folder+file, index=None)
```

Code listing B.2: Python script for resultant forces

```
import re
import os
import math
import numpy as np
import pandas as pd


if __name__ == '__main__':
    files = os.listdir('./data')

    pattern = re.compile(r'-?\d+\.*\d*')

    header = ['TopCurrent',
              'BottomCurrent',
              'Direction',
              'B1',
              'B2',
              'B3',
              'B4',
              'B5',
              'B6',
              'B7',
              'B8',
              'B9',
              'B10',
              'B11',
              'B12',
              'Net.Bottom.Pos2']

    df_save = pd.DataFrame(columns=header)

    for file in files:
        data_path = './data/' + file
        df = pd.read_csv(data_path, header=None, sep=',')

        forces_rawdata = df.iloc[5003:6003].values.astype(float)

        average_forces = np.average(forces_rawdata, axis=0)

        Net_Bottom_Pos2 = average_forces[0]

        B_forces = list()
        ii = 1
        for _ in range(12):
            i_force = math.sqrt(
                math.pow(average_forces[ii],
                2) + math.pow(average_forces[ii + 1],
                2) + math.pow(average_forces[ii + 2],
                2))
                B_forces.append(i_force)
                ii = ii + 3

        topcurrent = 111

        nums = re.findall(pattern, file)

        df_save = df_save.append({'TopCurrent': float(nums[0]),
```

```
                                'BottomCurrent': float(nums[1]),
                                'Direction': float(nums[2]),
                                'B1': B_forces[0],
                                'B2': B_forces[1],
                                'B3': B_forces[2],
                                'B4': B_forces[3],
                                'B5': B_forces[4],
                                'B6': B_forces[5],
                                'B7': B_forces[6],
                                'B8': B_forces[7],
                                'B9': B_forces[8],
                                'B10': B_forces[9],
                                'B11': B_forces[10],
                                'B12': B_forces[11],
                                'Net.Bottom.Pos2': Net_Bottom_Pos2},
                               ignore_index=True)

    df_save.to_csv('./samples.csv', index=None)
```

Code listing B.3: XGBoost python script

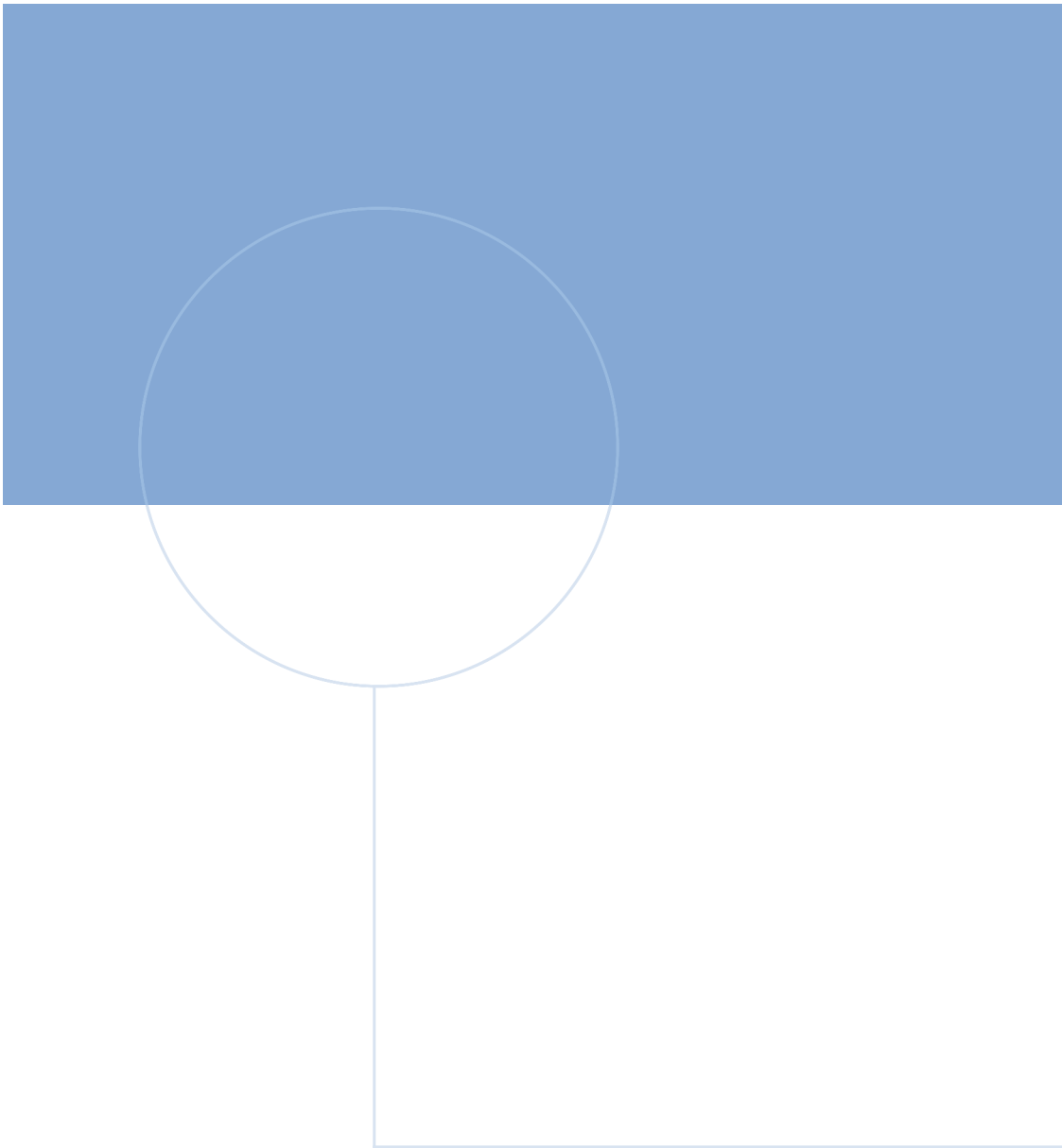
```
import math
import pandas as pd
import xgboost as xgb
from matplotlib import pyplot
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.model_selection import train_test_split


dataframe = df = pd.read_csv('./data/FULL_RESULTAT_ferdig.csv', sep=',')

data = dataframe.values
# split data into input and output columns
X = data[:, :3]


for i in range(3, 12+3):
    print('processing␣prediction␣{:2d}␣prediction'.format(i-2))
    y = data[:, i]

    # XGBoost training
    X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                        test_size=0.2,
                                                        random_state=0)

    model = xgb.XGBRegressor(n_estimators=600,
                             learning_rate=0.05,
                             min_child_weight=5,
                             eval_metric='rmse',
                             max_depth=5)

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)

    print("Force:␣{:2d},␣MAE:␣{:.2f}".format(i-2, mae))
    print("Force:␣{:2d},␣RMSE:␣{:.2f}".format(i-2, math.sqrt(mse)))

    df = pd.DataFrame(columns=['True', 'Prediction'])
    df['True'] = y_test
    df['Prediction'] = y_pred

    file_saving = 'pred_true_forces_' + str(i-2) + '.csv'

    df.to_csv(file_saving, index=False)
```

Appendix C

XGBoost results

Underneath is a hyperlink that is used as a forward to the XGBoost prediction data.

XGBoost result predictions forces.



NTNU

Norwegian University of
Science and Technology