Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Martin Ericsson

Fish-farm Integrated Sensor Cluster: Environmental and Biological Surveillance in Fish-farming Aquaculture with Emphasis on Sensor Fusion

Master's thesis in Electronic Systems Design
Supervisor: John R. Potter
Co-supervisor: Arild Søraunet

May 2022

NTNU
Norwegian University of
Science and Technology

Martin Ericsson

Fish-farm Integrated Sensor Cluster: Environmental and Biological Surveillance in Fish-farming Aquaculture with Emphasis on Sensor Fusion

Master’s thesis in Electronic Systems Design
Supervisor: John R. Potter
Co-supervisor: Arild Søraunet
May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Kunnskap for en bedre verden

Preface

This report is the end result of the course *"TFE4930 - Electronic Systems Design, Master's Thesis"*. The work was conducted during the final semester of the 2-year Electronic Systems Design Master of Science in Engineering programme at the Norwegian University of Science and Technology (NTNU), under the Smart Sensor Systems programme option.

I've gotta say.. what an experience these past years have been. I've learned so much, faced countless challenges, worked long days, and now ultimately reached the finish line. What you're about to read is without a doubt the pinnacle of my academic contributions, and will most likely remain so for a while now that my academic endeavors transition into an engineering career. I hope you find this thesis as interesting as it was for me to write.

Acknowledgements

I would first like to pat myself on the back for giving it my all when working on this thesis and project. Don't worry, I'll thank everyone else soon, but hear me out. I rarely stop to appreciate what I've accomplished when I overcome new obstacles. Instead, I just look at the next problem to solve and push through, knowing very well that I'll likely meet new frustrations and feelings of hopelessness. So this one's for me. All right, that's enough ego-boosting.

I would like to express my gratitude to all my colleagues at Norbit who helped me along the way. Everything from motivating words to explaining for the 50th time how various hydroacoustic-related phenomena work. In no specific order, thank you to Magnus Kofoed, Magnus Andersen, Tormod Vaule, Arne Solstad, Guttorm Lange, Tony Haugen, Kristin Valle, Ceyhun Ilguy and Arild Søraunet. Also, thank you to John R. Potter for your academic supervision.

I would also like to thank my dearest family for their continuous support throughout my studies. I know you have tried to understand what I've been working with on a deep theoretical level many times. I'm not saying that you haven't managed to do so, but you have really listened when I have tried to explain engineering-related topics that are vastly different from your medical expertise. You have heard me rant on and express my frustrations numerous times, helped me proofread this report, and always shown immense support. For this, I am forever grateful.

Martin Ericsson

Martin Ericsson
Trondheim, May 16, 2022

Abstract

Norway is the largest producer and exporter of farmed Atlantic salmon (*Salmo salar* L.) globally, but with a mortality rate of 15.5 % in 2021, the Norwegian Atlantic salmon fish-farming industry has a far way to go to be considered optimal. Environmental and biological surveillance in fish pens is challenging due to their size and the complexity of the underwater environment within them, and simple subsea cameras are still the standard surveillance method during daily operations. Hence, subjective assessments done by the fish farmers are still industry standard when evaluating the health and welfare of the fish. Many products to gain more control of fish farming daily operations exist on the market. Most of these products are either directed toward environmental monitoring or acquiring data on the fish. The latter type of product typically relies on advanced 3D camera systems and machine learning to detect salmon louse, wounds, and estimate biomass. Accordingly, these systems are relatively expensive.

In this thesis, a specially developed multi-sensor prototype is used to investigate an optical and acoustic sensor fusion approach to estimate the size and swimming speed of individual Atlantic salmon. To the author's knowledge, no other products on the market utilize optical and acoustic sensor fusion for a similar application. The prototype also includes an orientation sensor and an aquatic environment sensor measuring two vital aquatic parameters for Atlantic salmon (water Oxygen (O_2) content and temperature). The applied theoretical fundamentals within each field of science are presented to lay the basis for how the post-processing software was developed, emphasizing why specific approaches were used. The prototype system hardware was designed and constructed in the specialization project preceding the master's project but is presented as previous work to enable a better understanding of the system's functionality.

The developed software for the project is applied to data gathered during two full-scale field tests. A Deep learning-based Computer Vision model trained on custom data is used to optically detect (YOLOv4) and track (DeepSORT) fish, enabling fusion with hydroacoustic data. The O_2 and water temperature data are fused through simple mathematical functions based on optimal thresholds for Atlantic salmon to determine an objective aquatic environment quality score. Orientation data is used to determine which compass heading the system is inclined towards to estimate water current direction.

The water current direction estimates are deemed inconclusive due to erratic and/or erroneous headings, most likely caused by the specific placement of the system during data acquisition. During both field tests, the fused aquatic environment quality scores are poor when using strict requirements based on optimal thresholds. The optical and acoustic sensor fusion approach showed promising and realistic results but ultimately requires further development to become a viable solution for a future product on the market.

Sammendrag

Norge står for den største andelen produksjon og eksport av Atlantisk laks (*Salmo salar* L.) globalt, men med en dødelighetsrate på 15.5 % i 2021 så har den norske lakseoppdrettsnæring en lang vei å gå for å bli optimal. Miljø-basert og biologisk overvåkning i laksemerder er utfordrende gitt av deres størrelse og kompleksiteten av miljøet i dem, og enkle undervannskamera er industri-standarden for overvåkning under daglige operasjoner. Subjektive vurderinger utført av lakseoppdretterne er dermed fortsatt standard prosedyre for å bedømme fiskens generelle helse og velferd. Det finnes mange teknologiske løsninger på markedet som skal gi oppdretterne mere kontroll under daglige operasjoner. Flertallet av disse systemene er rettet mot å enten overvåke miljø-parameter eller innhente data på fisken. Produkter for det sistnevnte er ofte basert på avanserte 3D kamera-system og maskinlæring for å detektere lakselus, sår og estimere biomasse. Slike system er dermed relativt dyre.

Et eget-utviklet multisensorsystem blir i denne rapporten utnyttet til å utforske fusjon av optisk og akustisk data for å estimere størrelse og svømmehastighet på individuelle atlantisk laks. Så vidt forfatteren vet er det ingen andre produkter på markedet som utnytter en lignende sensorfusjon i et slikt bruksområde. Systemet inneholder også en sensor for å måle absolutt orientering og en sensor som måler to livsviktige miljø-parameter for atlantisk laks (vannets oksygeninnhold og temperatur). De anvendte teoretiske aspektene innenfor de respektive vitenskapsfeltene er presentert for å legge fundamentet for hvordan data-behandlingen er utviklet. Her blir det vektlagt hvorfor de spesifikke implementasjonene ble utnyttet. Prototypen i seg selv ble designet og konstruert i spesialiseringsprosjektet gjennomført før master-prosjektet, men er presentert som tidligere arbeid for å gi en bedre forståelse for hvordan systemet fungerer.

Den utviklede programvaren blir benyttet på data innhentet under to fullskala-tester. En dyp lærings-basert datasynsmodell trent på egen-generert data blir brukt for å detektere (YOLOv4) og tracke (DeepSORT) individuelle atlantisk laks, som gjør utnyttbar fusjon med akustisk data mulig. Oksygen-/ og temperatur-data blir fusjonert gjennom enkle matematiske funksjoner basert på optimale terskler for atlantisk laks for å få en objektiv numerisk kvalitet på det akvatiske miljøet. Data fra orienterings-sensoren blir brukt for å finne hvilken kompass-retning systemet heller imot for å estimere vannstrømsretning.

Estimatene på vannstrømsretning blir dømt inkonklusive på grunn av varierende og/eller feilaktig retning, mest sannsynlig forårsaket av den spesifikke plasseringen av systemet under testene. Under begge fullskala-testene er den resulterende kvaliteten på det akvatiske miljøet dårlig når strenge krav for optimalt miljø er utnyttet. Den optiske og akustiske sensorfusjonen viser lovende og realistiske resultat, men vil til syvende og sist kreve videreutvikling for å bli en egnet løsning i et sluttprodukt.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Understanding the relatively rudimentary technological standards in fish-farming aquaculture is one that may be surprising to those not familiar with this field. This industry is large globally and in most cases reaps vast annual profits (Moe et al., 2008), but still seems to be lacking when it comes to the availability of technological systems to objectively assist and optimize daily operations. Advanced decision support systems which use objective data readings (sensors) to assess the current operation status are generally not widespread in the fish-farming industry. When compared to such systems used in the oil and gas industry, the complex biological factor definitely complicates objective assessment from such systems. Therefore, many daily operation situations still heavily rely on subjective assessment based on live camera video feeds and, in *some* cases, individual sensor readings from the aquatic environment. The reason behind this fact mostly comes down to one main factor; the underwater environment within the fish pen is very complex. This complexity arises for three reasons: the large aquatic volume, the large number of individuals contained within the volume, and that it's underwater. Nevertheless, research within this field is very active and must continue in the future to converge the industry standards towards a truly sustainable and optimal level.

In the following, the current state and desired future of Norwegian fish farming with its challenges are presented. The available technological solutions targeted for this industry, which are relevant to the scope of this thesis, are then briefly discussed before presenting the approach focused on in this report.

1.1 Background and Motivation

Norwegian fish-farming aquaculture has seen vast growth since its beginnings in the 1970s and Norway has become the largest producer and exporter of farmed Atlantic salmon (*Salmo* *salar* L.) globally (Bailey et al., 2020). The demands for sustainable aquatic protein sources are increasing and it is deemed that fish-farming aquaculture is one of the main contributors to meeting these future needs (FAO, 2018). This requires a continuation of production up-scaling in the industry, but this up-scaling could also increase the likelihood of facing emerging social, biological, and economical challenges (Føre, Frank, et al., 2018).

In Norwegian fish farms, a typical fish pen holds up to 200 000 individuals, has a circumference of up to 157 m, and contains a volume of approximately 40 000 m^3. Many fish-farming sites operate 8-16 fish pens, resulting in the responsibility for several millions of individuals simultaneously (Føre, Frank, et al., 2018). This is obviously a large and very complex environment to monitor effectively, considering the large total aquatic volume. Individual-based relationships with the animals, which is comparatively easy in livestock farming with much fewer individuals, is not possible in this scale of fish farming given the number of individuals within each pen. Instead, underwater cameras are typically installed in every fish pen to enable the fish farmers to monitor the underwater environment and use subjective assessment to draw conclusions based on observed fish behavior. In reality, the cameras observe a very small percentage of the total volume, but this is still deemed adequate for total status determination.

Biological/physical metrics on the fish and measurements from the aquatic environment itself can be acquired by using optical, acoustic, and environmental sensor systems. Since this thesis is concentrated on farming of Atlantic salmon, no other species of fish are covered in the report. There are a variety of products on the market that are targeted toward measuring aquatic environmental parameters which are important for Atlantic salmon health and welfare, such as Oxygen (O_2) content, temperature, salinity, water current speed, and more[1]. For measuring biological metrics, such as individual size, lice-counting, and depth-distribution within the pen, a variety of hydroacoustic and camera-based solutions exist. These types of systems generally give the fish farmer more data on the state within the fish pen, assuming that the farmer knows what is normal or best for the fish. However, since many of these products are observational systems, based on single-parameter units (i.e. only outputs one parameter on a display), the determination of the total situation status within the pen relies on the subjective assessment done by the fish farmer based on prior experience. This has historically been, and still currently is, the industry standard (Føre, Frank, et al., 2018).

Nevertheless, several vendors have developed state-of-the-art systems targeted toward alleviating the fish farmers from performing some of these subjective assessments and minimizing excessive manual labor. A few selected vendors which are at the forefront of these types of solutions are presented below.

[1] https://www.akvagroup.com/sj%c3%b8basert-oppdrett/kamera-sensorikk/milj%c3%b8sensorikk

Stingray Marine Solutions[2] have developed a stereo camera system with an integrated laser that fires pulses at detected louse on Atlantic salmon. The louse-detection is based on Computer Vision and machine learning. This product serves as an autonomous de-lousing system that ultimately improves fish health and welfare, as well as helps decrease financial losses related to lice infestations. Although lice infestations are not considered in this thesis, it is important to mention them since this is a recurring issue in the industry that affects health and welfare.

Aquabyte[3] deliver a fish welfare determination software solution based on machine learning, where the main focus is directed towards autonomous counting of lice and estimating fish biomass. The welfare determination seems to be based on observing trends in total lice count. They are partnered with *Imenco*[4], which deliver a specifically developed stereo camera which interfaces to *Aquabytes* software. *Imenco* themselves additionally supply aquaculture-targeted surveillance camera solutions and have developed pellet detection software that can be used to minimize feed loss.

Optoscale[5] have developed an advanced camera system with incorporated machine learning-based software. Their camera uses structured light to generate estimates of individual fish weight with up to 98 % accuracy. This allows fish farmers to have precise control over their biomass, which is important for maximizing profits. *Optoscale* additionally delivers welfare determination and lice-counting based on the same camera platform. The welfare determination is based on the detection of lice, wounds, deformations, and sexual maturation.

The systems mentioned above, and other similar products on market, are highly technologically advanced and function well enough to be desirable for the aquaculture industry. Most of them seem to mainly rely on high-resolution 3D camera imaging, i.e. expensive camera systems, to assess fish welfare. Some of these products also implement aquatic environmental sensors as an add-on to the camera housing. However, generally very few products implement a handful of different sensors in the same housing to collect more parameters, which is important for total status determination within the fish pen.

As will become apparent shortly, this thesis focuses on the application of aquatic environmental sensors, absolute orientation information, and a customized Deep learning pipeline to achieve optical and acoustic sensor fusion. To the author's knowledge, no other products on the market targeted for fish-farming aquaculture have a similar mix of integrated sensors in the same housing.

[2] `https://www.stingray.no/`
[3] `https://www.aquabyte.no`
[4] `https://imenco.no/_/aquaculture`
[5] `https://optoscale.no`

The main goal of the thesis is to utilize the prototype and developed post-processing software to investigate the feasibility of a system comprising these sensors. This is investigated by applying the following processing to the implemented sensors:

- O_2 and Water Temperature Fusion: To determine the quality of the aquatic environment.
- Optical and Acoustic Fusion: To estimate the size and swimming speed of individual Atlantic salmon through the application of Deep learning.
- Water current direction estimation through system orientation information.

1.2 Report Outline

Chapter 2 outlines the fundamental theoretical aspects applied in the development of the systems post-processing software. Here you will find a presentation of basic hydroacoustic signal processing, orientation-sensors, monocular camera geometry, applied optical and acoustic sensor fusion approach, relevant Atlantic salmon welfare parameters, and lastly the implemented Deep learning pipeline theory.

Chapter 3 presents the Fish-farm Integrated Sensor Cluster (FISC) prototype. Here you will find a brief presentation of the (prior) specifically designed hardware, as well as a full walkthrough of the custom-developed software with its implementation frameworks and method. Additionally, the details on two full-scale field tests performed during the project duration are presented.

Chapter 4 presents a selection of obtained results during the project. These include the vertical characterization of the system's hydroacoustics, a presentation of data acquired from the aforementioned field tests, and the results from the implemented post-processing and sensor fusion.

Chapter 5 analyzes the obtained results from the previous chapter, where a more critical view of the feasibility of the system in its current state is discussed.

Chapter 6 briefly summarizes and concludes the thesis.

Chapter 2

Theory

This chapter outlines the theoretical fundamentals which have been applied in the work presented in this report. The hardware used in the project was designed and developed in the semester preceding the master's thesis, where the theory related to design aspects and initial system tests were documented in the specialization project report (see (Ericsson, 2021)). Consequently, this report only covers work performed between January and May 2022. The implemented system and project in general covers several quite different theoretical fields. Since the desired result was to achieve a functional prototype for future product development, the theoretical depth within each of these fields could be somewhat trivial for a knowledgeable reader within the respective fields. Nevertheless, the fundamentals behind the chosen processing procedures are presented in such a way that, hopefully, readers with little to no prior experience within these fields can understand how and why the applied approaches are used.

2.1 Hydroacoustic Signal Processing

Active hydroacoustic systems, such as sonars and echosounders, operate by transmitting acoustic pulses and use the received echoes to detect targets. The distance to the targets are determined from the following equation (Hovem, 2012, p.277):

$$d = \frac{c}{2}\Delta T \tag{2.1}$$

where
c = Sound speed in water [m/s].
ΔT = Time between pulse transmission and echo reception [s].

The transmitted pulses have pre-determined parameters, such as center frequency, pulse length, and bandwidth. The receiver can therefore be designed to optimally detect signals with the known characteristics of the transmitted signal through *matched filter processing*. Matched filters are linear time-invariant (LTI) filters that perform a cross-correlation of the received echo-data with a time-reversed (conjugated) replica of the transmitted signal. Hence, they are also known as replica correlators. Matched filters are optimal for Signal-to-Noise Ratio (SNR) maximization to detect known signals in additive white Gaussian noise, and are therefore very common in radar and sonar systems (Lyons, 2011, p. 469; Waite, 2001, pp. 163-164; Mahafza et al., 2021, p. 183).

In the following, a bit of theory on how matched filters function and a few methods of implementation are detailed. For a more extensive and theoretical presentation of matched filters, the reader is referenced to Mahafza et al. (2021, pp. 183-229).

For time-continuous signals, matched filtering is performed through a convolution integral between the time-reversed pulse replica and the input signal. The input signal includes returns of the transmitted signal (echoes) and noise (Mahafza et al., 2021, pp. 183-188). In discrete (digital) matched filters, Finite Impulse Response (FIR) filters are often used. The FIR filter's impulse response should then equal the discrete reversed pulse replica. The correlation operation, similar to convolution, is achieved through a multiplication, shift, and summation throughout the signal, i.e. (Brigham, 1988, p. 272):

$$y[i] = \sum_{i=0}^{N-1} h(n)x(n-i)$$

where
$h(n)$ = FIR filter impulse response.
$x(n)$ = Discrete input signal with echoes.

Although this method is easy to implement and achieves the desired results in terms of the optimal SNR-enhancing properties of matched filters, it is obvious that the computational efficiency is highly governed by the total amount of samples, and could be slow when N is large. The utilization of frequency-domain computation through a Fourier transform could increase the computation efficiency in a replica correlator. The Fourier transform decomposes a time-domain signal to its frequency content and is extensively used in most signal processing fields. In many programming libraries, such as the Scipy and Numpy package for Python, the Cooley-Tukey Fast Fourier Transform (FFT) algorithm is commonly used for time-to-frequency domain transformations due to its low computation time (Brigham, 1988, p. 131).

The discrete convolution theorem states that discrete time-domain convolution, and hence cross-correlation as well, is equal to frequency-domain multiplication (Brigham, 1988, p. 112).

Therefore, it could be beneficial to perform an FFT of both the matched filter impulse response and the filter signal input, multiply them, and lastly compute the inverse FFT to achieve the time-domain cross-correlated result. Although the time-to-frequency domain transformation steps might intuitively seem inefficient, this procedure is in most cases more efficient than the direct time-domain cross-correlation. For this reason, this procedure is said to be "*a shortcut by the long way around*" (Brigham, 1988, p. 207). Implementations of cross-correlation in Python libraries often let the user choose between the direct or FFT-based method.

Since the cross-correlation process *in essence* consists of shifting the pulse replica across the receiver input signal and multiplying at each step, the filter output signal will reach its maximum when the echo pulse and replica pulse overlap. Due to this temporal coherence, the filter output is applicable for determining the distance to detected targets. It is the envelope of the resulting correlation output that is directly applicable for basic target detection, and all hydroacoustic signal processing in this report focuses on the extracted envelope. A matched filter correlation is demonstrated in Figure 2.1.

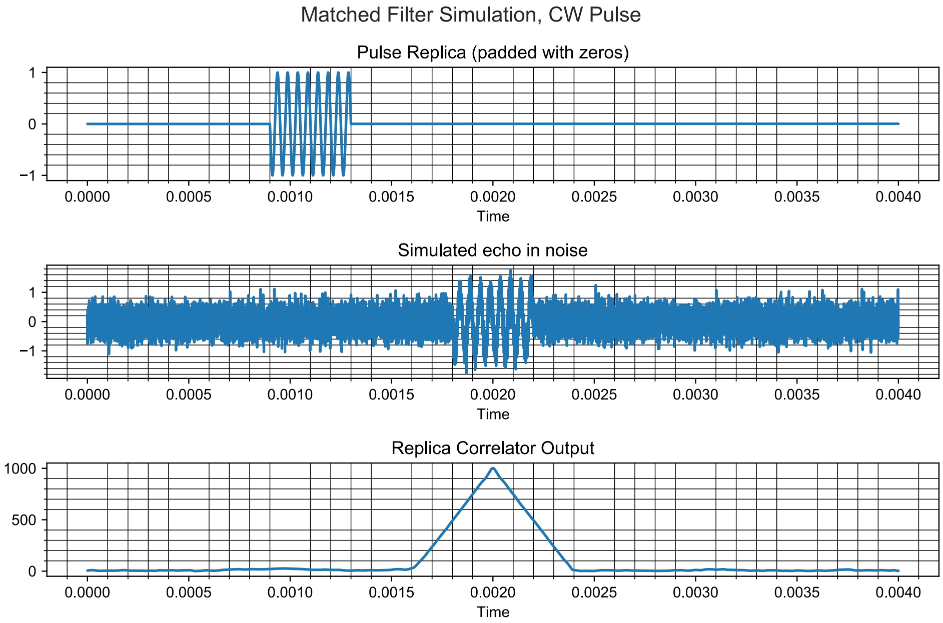


Figure 2.1: Plot showing replica correlation.

2.1.1 Pulse Compression

When cross-correlating a transmission pulse without any frequency bandwidth, i.e. a sinusoidal Continuous Wave (CW) pulse, the temporal duration of the replica correlator's pulse overlap is dependent on the pulse length (T_p). This is observed in the last plot in Figure 2.1. In this case, the duration of the pulse governs the range resolution of the system, since two targets close to each other in the propagation direction might be difficult to individually detect. The range resolution (Δr_{CW}) in a system using a CW pulse with two-way transmission is approximated with the following equation (Hovem, 2012, p. 303; Mahafza et al., 2021, pp. 191-193):

$$\Delta r_{CW} = \frac{c}{2} T_p$$

where
c = Sound speed in water [m/s].
T_p = Pulse length [s].

When several targets are in close proximity to each other in the propagation direction, a high range resolution (low Δr) is necessary to acoustically differentiate individuals. The trade-off with this fact is that the SNR decreases with the pulse length due to the decrease in average transmitted power (Mahafza et al., 2021, p. 204).

Pulse compression is a matched filter technique used in most radar and sonar systems that further exploit the characteristics of replica correlation. By using a coded transmission, the range resolution can be increased significantly due to the temporal width being compressed after correlation (Abraham, 2017). There are a variety of pulse coding schemes, where normally the frequency or phase changes during the pulse duration. The simplest coding scheme to implement is a Linear Frequency Modulation (LFM) pulse, which linearly sweeps across a bandwidth during the pulse duration. The compression occurs due to the intra-pulse variations, meaning the correlation output will rapidly increase when the replica and echo pulse approach a perfect overlap, as shown in Figure 2.2. This focuses the energy in the center of the pulse and the LFM range resolution (Δr_{LFM}) is then mainly dependent on the bandwidth of the pulse, i.e. (Hovem, 2012, p. 277):

$$\Delta r_{LFM} \approx \frac{c}{2BW}$$

where
c = Sound speed in water [m/s].
BW = Pulse frequency bandwidth [Hz].

With pulse compression, it is possible to achieve a high range resolution while maximizing the SNR by utilizing a transmission pulse that has a long time duration (high energy) and a bandwidth that is much larger than $1/T_p$ (large compression ratio). A side effect of LFM pulse compression is the temporal sidelobes present around the main lobe (pulse center), as shown in Figure 2.2. This is easier to observe when plotting the instantaneous power of the correlation output, demonstrated in Figure 2.3.

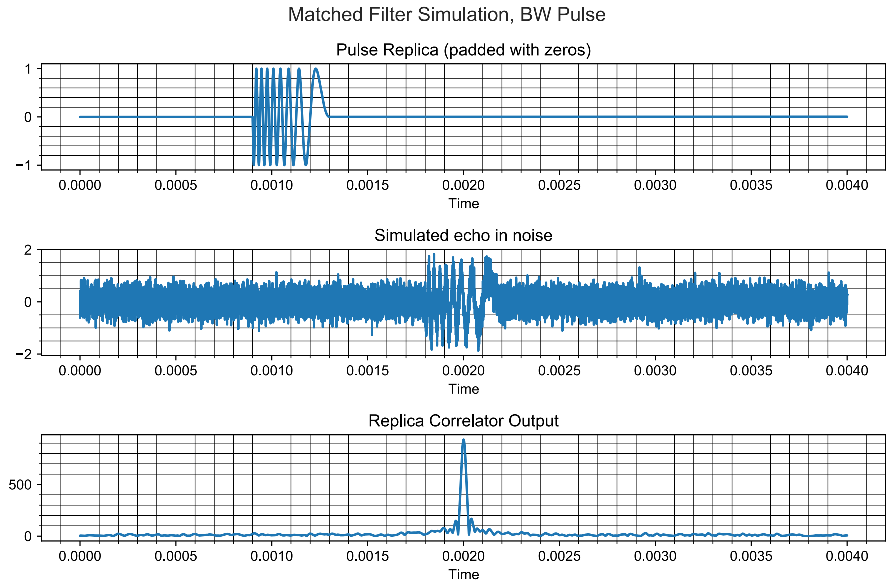


Figure 2.2: Plot showing example of LFM pulse compression.

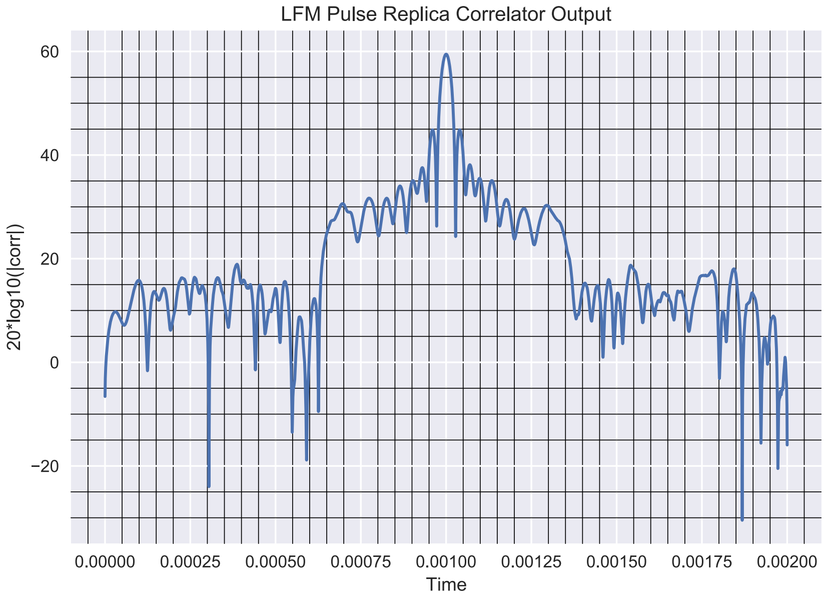


Figure 2.3: Instantaneous power output of LFM pulse compression example.

When the pulse is given a frequency bandwidth, the spectral content will include a rectangular ”window” over the set bandwidth. Since the FFT of a rectangular window in the time domain produces a sinc-pulse shape in the frequency domain, and duality is inherent in Fourier transforms, the inverse FFT of a rectangular window will exert a sinc-pulse shape in the time domain (Brigham, 1988, pp. 101-103; Mahafza et al., 2021, pp. 40-41; Abraham, 2017).

The sinc-pulse temporal sidelobes are undesirable since they could be interpreted as detections of small targets before and/or after the actual target by the detector. To minimize sidelobe levels, windowing is the easiest and best method to utilize. Windowing is the general term for truncating data or subsets of data, where the window is zero-valued outside the region of interest and weighted inside the window. In hydroacoustic signal processing, the Hamming window is commonly applied.

The Hamming window tapers the amplitude of the signal on both sides to nearly zero, such that there are no rapid changes in amplitude, unlike the rectangular window. In the frequency domain, the rectangular shape of the bandwidth is rounded as a result, since the spectral energy is decreased in the lower and higher frequencies within the bandwidth. This decreases the first sidelobe level to approximately 40 dB below the main lobe peak. It is important to note that both the width and peak level of the main lobe is affected by windowing. For the Hamming window, the main lobe peak reduction factor is 0.73, and the main lobe width is doubled. (Mahafza et al., 2021, p. 44). A comparison between the rectangular and Hamming window on the same pulse is shown in Figure 2.4.

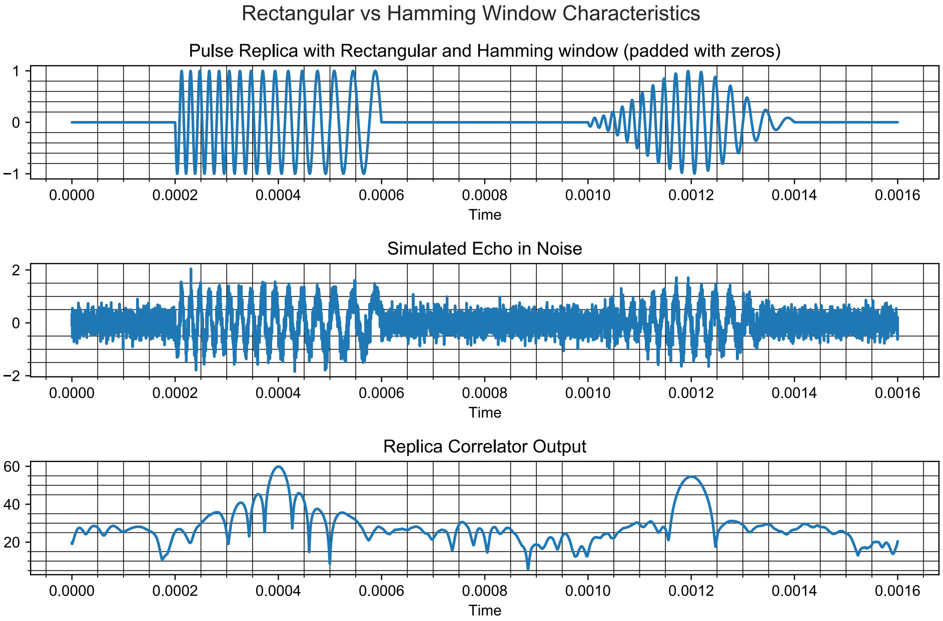


Figure 2.4: Peak-reduction and mainlobe width trade-off, rectangular versus Hamming window.

2.1.2 Target Detection

When applying a matched filter receiver, the direct approach to target detection is a peak detector. Detection of peaks in noise is a binary hypothesis test since there are two possible conclusions, i.e.:

- Decide H_0 = No Target Present (null hypothesis)
- Decide H_1 = Target Present

There are four possible scenarios based on these two conclusions, namely:

1. P_D = Probability of Detection
    - Correct conclusion of target present $P(H_1; H_1)$
2. P_R = Probability of Rejection
    - Correct conclusion of no target present $P(H_0; H_0)$
3. P_{FA} = Probability of False Alarm
    - Wrong conclusion of target present $P(H_1; H_0)$
4. P_M = Probability of Miss
    - Wrong conclusion of no target present $P(H_0; H_1)$

The probabilistic theory behind hypothesis testing is not elaborated further, but the fundamentals above are simply introduced to accompany the following paragraphs.

The required signal level to decide H_1 is called the detection threshold. With a threshold far too low, the number of false alarms will be large ($P_{FA} \uparrow$). Conversely, with a threshold too high, several present targets will be missed due to the increased SNR demand for detection to occur ($P_D \downarrow$). This denotes a trade-off that peak-detectors should ideally optimize. There are a large variety of peak-detection algorithms used in radar and sonar applications, but one specific algorithm was implemented on recommendation from an engineer at Norbit Subsea.

The Constant False Alarm Rate (CFAR) algorithm is a multiple-target detector that adaptively alters the detection threshold during the processing of an echo-return. Although there are several variations of CFAR detectors, the Cell Averaging CFAR (CA-CFAR) is the most common and is easy to implement in software. As the name suggests, the detection threshold is adapted such that the probability of false alarm is held constant. The following presentation is inspired by Holm (2010, pp. 589–607).

The CA-CFAR algorithm shifts throughout the echo-data and performs a hypothesis test for every sample (range cell). The current range cell undergoing the hypothesis test is named the

Cell Under Test (CUT). The cell structure is illustrated in Figure 2.5.

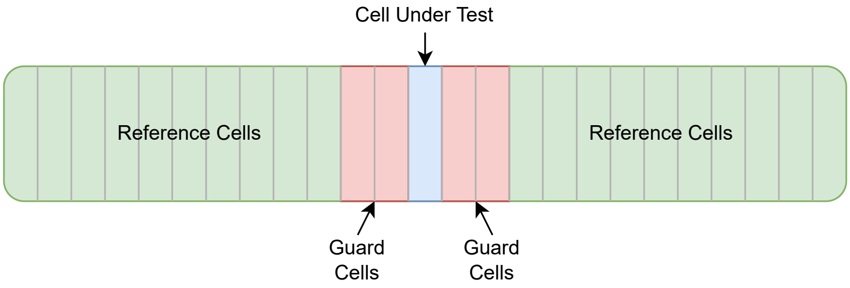


Figure 2.5: CA-CFAR cell structure illustration.

The adaptive threshold in every step is based on 1) a scaling factor and 2) an estimate of the background noise in the preceding and succeeding reference cells, excluding the guard cells. The scaling factor, α, is constant and is given by the following equation:

$$\alpha = N(P_{FA}^{-\frac{1}{N}} - 1) \tag{2.2}$$

where
N = Number of reference cells.
P_{FA} = Chosen probability of false alarm.

The background noise is assumed to be heterogeneous over the entire window and is hence assumed to be a valid estimate of the noise in the CUT. The noise is found with the following equation:

$$\text{Noise} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2.3}$$

where
N = Number of reference cells
x_n = Value of reference cell n.

If the CUT is in fact the peak of a signal, the guard cells should avoid wide target echoes from corrupting the background noise estimate. The adaptive detection threshold is computed based on the estimated background noise and the aforementioned scaling factor, i.e.:

$$T = \alpha \cdot Noise \tag{2.4}$$

If the CUT has a value larger than the current threshold (T), a target is assumed to be present and the algorithm outputs a confirmed detection.

2.2 Absolute Orientation Determination

Inertial Measurement Units (IMUs) are devices used for measuring the three-dimensional (3D) motion and orientation of a body, where linear forces, angular rotation and sometimes heading direction is determined. These parameters are measured by using accelerometers, gyroscopes, and magnetometers, respectively. In an absolute orientation sensor, there are three of each sensor type mounted orthogonal to each other to produce measurements in three perpendicular directions, i.e. three dimensions (Shkel, 2021, pp. 2-3). By utilizing the data from three of each sensor mounted in this way, the result is an IMU with nine degrees of freedom (9DOF), resulting in absolute 3D orientation estimation. Such sensor packages are also often called Attitude and Heading Reference Systems (AHRS).

The technological advancements of Microelectromechanical Systems (MEMS) since their origin in the late 1970s (Mohinder S. Grewal, 2007, p. 331) have resulted in IMUs becoming increasingly accurate and very compact, reaching down to just a few millimeters in size with capabilities of measuring complete 3D orientation (Shkel, 2021, pp. 5-6).

Typically, such MEMS-based IMUs can output individual sensor readings or even absolute orientation information directly through e.g. a serial interface. This allows for a simple integration interface with a Microcontroller Unit (MCU). Since the BNO055 is used in this project, and it can output absolute orientation estimates directly, the following paragraphs assume similar sensor integrations.

Orientation in 3D is easiest to directly interpret by using Euler angle notations referenced to a fixed coordinate system. With this representation, rotations around the x, y, and z-axis are often named roll, pitch, and yaw, respectively. Another terminology for yaw is heading, since it describes the horizontal compass direction. Figure 2.6 visualizes roll, pitch, and yaw in a fixed coordinate system.

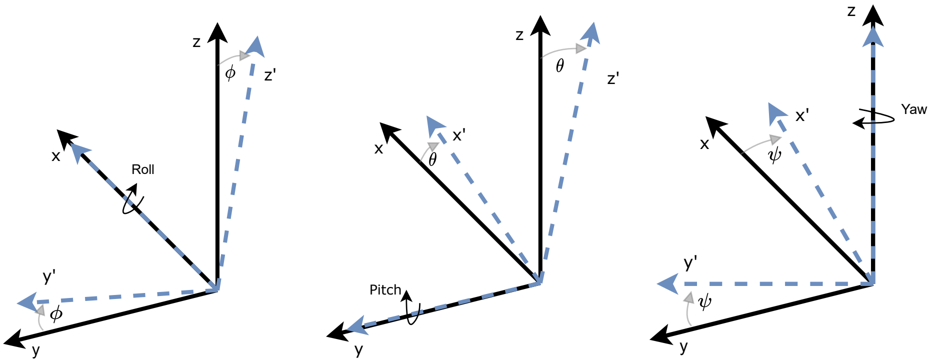


Figure 2.6: Visualization of roll, pitch and yaw.

Although describing the orientation with Euler angles is intuitive, the operation of rotating between two different orientations, using Euler angles, involves three sequential rotations about the three axes in the coordinate system. This procedure can lead to gimbal lock, where one degree of freedom is lost, and also requires a 3x3 matrix to be stored in memory before computation of the new orientation. Another representation of orientation in 3D space that does not suffer from gimbal lock, and is also more computationally efficient, are unit quaternions (Titterton et al., 2004, pp. 36-47; Kok et al., 2017, p. 19).

Quaternions are composited as a 4-dimensional complex number, which has one real component and three imaginary ones. Including the definition that the sum of its squares must be equal to one, this 4D vector is a unit quaternion, i.e. (Titterton et al., 2004, p. 43; Kok et al., 2017, p. 20):

$$\begin{aligned}
&\mathbf{q} = a + \mathbf{i}b + \mathbf{j}c + \mathbf{k}d = q_0 + q_1 + q_2 + q_3 \\
&\mathbf{q} \in \mathbb{R}^4 \quad ||\mathbf{q}|| = 1 \\
&\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1
\end{aligned} \tag{2.5}$$

Since the orientation information is embedded in the algebraic values of the quaternion coefficients, less memory is needed and only multiplication and addition is necessary to compute orientation changes. This is also the reason why several MEMS-based orientation sensors have the option to directly output quaternions, such as the BNO055. For a comprehensive derivation of quaternions and their qualities, the reader is referred to see the work by Hanson (2006). Since orientation is difficult to directly interpret from quaternions, it is often desirable to convert them to Euler angles to decompose the orientation to the aforementioned roll, pitch, and heading convention. This conversion is done with the following equations (Blanco, 2010, p. 15):

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} Roll \\ Pitch \\ Heading \end{bmatrix} = \begin{bmatrix} arctan(\frac{2(q_0q_1+q_2q_3)}{1-2({q_1}^2+{q_2}^2)}) \\ arcsin(2(q_0q_2 - q_1q_3)) \\ arctan(\frac{2(q_0q_3+q_1q_2)}{1-2({q_2}^2+{q_3}^2)}) \end{bmatrix} \tag{2.6}$$

2.3 Subsea Cameras in Aquaculture

As mentioned in the introduction, subsea cameras are the traditional method of fish-pen surveillance in aquaculture. Each fish pen often has several static mounted cameras for manual operation, where fish farmers mostly use subjective assessment to quantify fish behavior. This essentially results in daily operations, such as feeding duration, feeding intervals, and general fish-pen state determination relying on the experience of each individual fish farmer who is responsible at the time.

In addition to these simple cameras, there are a variety of highly advanced camera systems for automatic biomass estimation[1][2], lice detection and welfare determination[3] and possibly in the future, individual fish identification through biometrics[4]. Systems like these were detailed in Section 1.1, but simply re-iterated here to refresh your memory. Cameras used for these purposes mostly utilize more than one sensor unit to get the visual data, through e.g. a structured light module or by using two (or more) cameras. This results in direct depth knowledge in the resulting image(s), from which 3D information can be extracted. These approaches are used in many industrial products and have solid track records when it comes to accuracy and reputation. In this project, a singular low-cost camera and a single-beam echosounder sensor fusion approach is investigated. Before this approach can be explained, some geometrical qualities of cameras must be understood.

2.3.1 Monocular Camera Geometry

When a monocular camera system is used for object detection with determination of 3D geometry, e.g. object size, the distance to objects must also be known. Without knowing this distance, 3D geometry can not be extracted from the pure 2D information given from the camera[5].

In addition to depth information from another sensor, the Field of View (FOV) is an intrinsic camera parameter that must be determined to estimate the physical scale of objects within the frame. The FOV describes the visible angular limits in the horizontal and vertical direction, giving information on the maximum area that the camera can observe at a given distance from it. The FOV is given from the physical sensor size and the lenses focal length. The focal length is the distance from the sensor surface to the optical center of the lens (Kruegle et al., 2006).

Normally, these parameters are given in the datasheet of the camera/lens module, but these values apply when the camera/lens is used ”as intended” without any optical modifications. When submerging a camera in water with an additional water-tightening lens, hereby simply referred to as lens, the FOV will decrease due to the additional lens becoming an interface between the water and the air inside the housing. Due to water having a larger refraction index than air, the incident light will be bent towards the normal through the lens, resulting in the FOV decreasing through the water-air interface. This effect arises from Snell’s Law of Refraction. Conversely, the focal length is increased in this situation (Menna et al., 2016) and is shown in Figure 2.7.

[1] https://optoscale.no/

[2] https://www.sealab.no/

[3] https://www.aquabyte.no/

[4] https://www.biosort.no/

[5] There are however machine learning-based approaches to determine depth information from training deep neural networks (Chanduri et al., 2021; Godard et al., 2018), but this is out of the scope of this thesis.

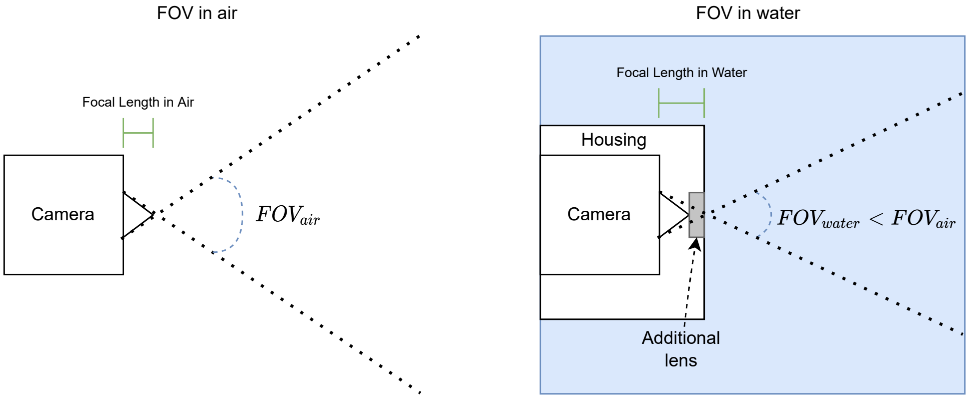


Figure 2.7: Illustration of FOV decrease caused by a water-air interface.

Characterizing a monocular camera to find its horizontal and vertical FOV is a process that can be broken down into simple trigonometry. Figure 2.8 illustrates this characterization setup, where the necessary distances are easy to measure manually.

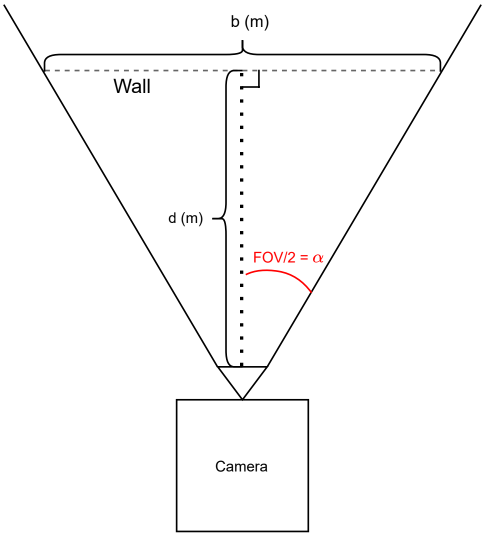


Figure 2.8: Monocular camera FOV characterization setup.

The field of view in either axis is then calculated by using the following equation:

$$FOV_x = 2\alpha = 2arctan(\frac{b_x/2}{d}) \tag{2.7}$$

where

x = Horizontal or vertical.

b_x and d are measured manually with same unit (m/cm/mm).

It is assumed that the camera module is installed perfectly aligned with the system orientation-sensor in all axes of rotation. Furthermore, the pincushion distortions caused by the flat water-tightening lens, which result in the outer edges of the visible frame bending inwards (see (Menna et al., 2016) for more), are ignored. This will be elaborated further in Section 5.4.

2.4 Optical and Acoustic Sensor Fusion Approach

The application of optical and acoustic sensor fusion can exploit the respective intrinsic advantages of both sensors to extract more information than they are capable of individually. On a superficial level, optical sensors have superior resolution while hydroacoustic sensors have far better range. Ferreira et al. (2016) presents several state-of-the-art approaches to this type of sensor fusion, where the inherent benefits, applications, and challenges are detailed through presenting prior research. Since the main focus of the paper is directed toward state-of-the-art applications, most of the cited work implements the fusion on a scientific level which is beyond the scope of this thesis. These implementations include systems that use advanced multibeam sonar systems to acquire acoustic data. Furthermore, the acoustic data in most of the cited work is used for improving underwater imaging applications. Nevertheless, a few of the cited papers utilize a similar sensor combination for fusion, namely a single-beam echosounder and a monocular camera.

Assuming that the camera's horizontal and vertical FOV, as well as the radial distance to an object is known, it is possible to decompose image pixel coordinates to 3D points in space relative to the camera. In this thesis, the aforementioned superficial advantages of neither sensor are exploited. Instead, the depth information from the acoustic sensor is fused with data extracted from images.

When single-beam hydroacoustics are used to find the distance to targets, the echosounder will yield the *radial* distance. Figure 2.9 shows the decomposition of an object's position with radial distance, d, to the distances in each axis, as well as the reference to the camera frame. For simplicity, it is assumed that the echosounder is a point source located in the camera center. To avoid confusion, note that the orientation-related axes in the preceding section were defined differently, where z was the vertical axis and x was the forward axis. For camera-related geometry, the notations shown in Figure 2.9 are used instead.

The following calculations are based on breaking down the information from the intrinsic camera FOV and object pixel coordinates into angles. The two angles to be determined are the angles formed between the center of the frame to the x_{px} and y_{px} coordinates of the object, which only relies on information from the camera frame. These two angles are θ and ϕ, which together with the radial distance from the echosounder, can be used to estimate the relative 3D Cartesian coordinates for optically and acoustically detected objects.

Figure 2.10 shows an example situation where the unknown angle θ needs to be determined.

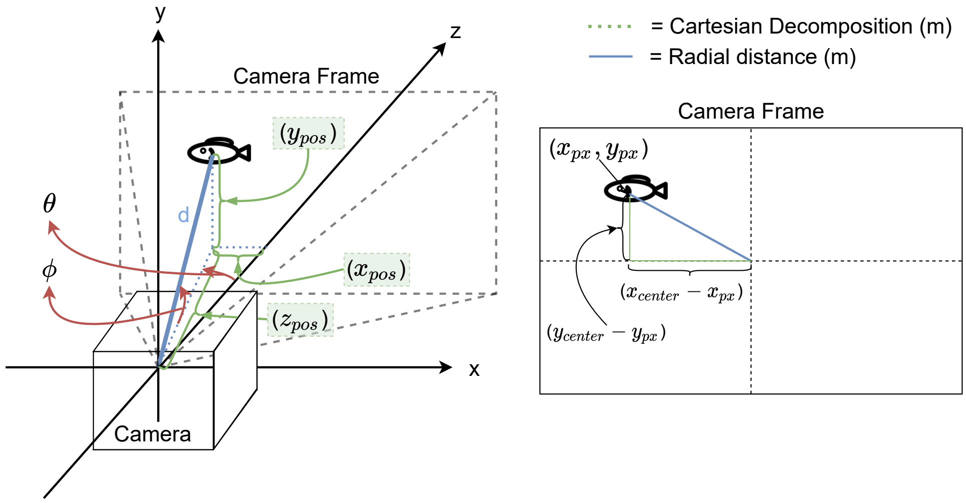


Figure 2.9: Decomposition of object position from radial distance and frame location.

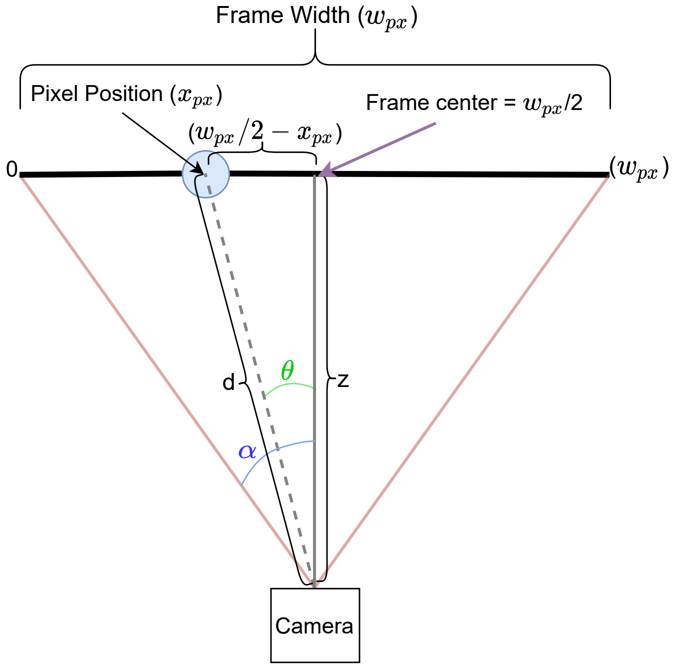


Figure 2.10: Horizontal pixel coordinate to angle illustration.

To determine an equation for θ, we first apply simple trigonometric equations to the two triangles given by θ and α and solve them for z, i.e.:

$$tan(\alpha) = \frac{w_{px}/2}{z} \qquad tan(\theta) = \frac{w_{px}/2 - x_{px}}{z}$$

$$\Downarrow \qquad \Downarrow$$

$$z = \frac{w_{px}/2}{tan(\alpha)} \qquad z = \frac{w_{px}/2 - x_{px}}{tan(\theta)}$$

Now that we have two equations for z, we can equate them and solve for θ:

$$\frac{w_{px}/2}{tan(\alpha)} \qquad \Longleftrightarrow \qquad \frac{w_{px}/2 - x_{px}}{tan(\theta)}$$

$$\theta = arctan(\frac{(w_{px}/2 - x_{px})tan(\alpha)}{w_{px}/2}) \tag{2.8}$$

where
w_{px} = Frame width in pixels.
x_{px} = Horizontal pixel coordinate of object in frame.
α = Half of horizontal FOV.

Note that neither z or d from Figure 2.10 are needed to find these angles, meaning θ is extracted purely from 2D image information.

Using the equations above with the vertical FOV, frame *height* in pixels and the vertical pixel coordinate of the object, will yield the angle between the frame center and the vertical offset of the object (ϕ). With these two angles and the radial distance, the relative Cartesian x and y coordinates for the object, in meters, are found with the following equations:

$$x_{coord} = dsin(\theta) \tag{2.9}$$

$$y_{coord} = dsin(\phi) \tag{2.10}$$

where
d = Radial distance to object from echosounder [m].

Finding the true z-distance to the object can now be solved by applying the Pythagorean theorem, since it scales to any dimension, i.e.:

$$d^2 = x^2_{coord} + y^2_{coord} + z^2_{coord}$$

$$z_{coord} = \sqrt{d^2 - x^2_{coord} - y^2_{coord}} \tag{2.11}$$

These Cartesian coordinates can now be used to determine object scale based on its pixel size in the image frame, as well as estimate object velocities when the time duration between two detections of the same individual is known. However, this requires the objects to be detected optically and acoustically, as well as matched across the two separate detections spaces. This will be detailed later.

> *For readers not intrigued by the theoretical and engineering-related topics up until this point, you might be delighted to know that you will now get a brief hiatus from engineering by taking a small detour within the field of biology. If this is the case, the bad news for you is that we will later return to deeper levels of theoretical engineering topics. (Yes, this is a case of foreshadowing.)*

2.5 Atlantic Salmon Welfare Indicators

The focus on the health and welfare of farmed animals has seen a general increase over the past decades. In the latest years, fish-farming aquaculture has been socially scrutinized for in some aspects failing to meet fish-specific welfare needs to the same extent that other farmed animals receive, where many occurrences of stress-/ or disease-induced mortality have been recorded (Sommerset, Jensen, et al., 2021). In 2021, an all-time high of 54 million Atlantic salmon died during the on-growing phase in sea-pens in Norway alone. This constitutes a mortality rate of 15.5 %. (Sommerset, Walde, et al., 2022, p. 6).

The Norwegian regulations specified for fish-farming aquaculture (Lovdata, 2008) are rather complex. When compared to other live-stock regulations, such as those relevant to chicken farming, it is found that the Norwegian aquaculture regulations unintentionally downgrade the importance of meeting the health and welfare needs of fish. In essence, farming of fish in Norway has less strict demands for welfare and mortality control when compared to other live-stock (Gismervik et al., 2020).

Nevertheless, a substantial amount of research and industry-specific official guidelines have been published to characterize and assess optimum welfare parameters for Atlantic salmon, where both environmental and biological factors are detailed. Several of these parameters are found in the regulations but are in some cases rather vaguely defined. In the Norwegian aquaculture regulations (Lovdata, 2008), §22 states that the water amount, water quality, water flow, and water current speeds should be at a level that ensures good welfare. The same paragraph also states that the waters O_2 content, temperature, and ”other parameters which are of considerable importance to fish welfare” should be measured systematically. The parameters included in this thesis are presented below.

2.5.1 Swimming Speed

The maximum sustainable swimming speed (U_{crit}) for Atlantic salmon has in the recent years been studied due to the increasing desire to move fish-farming aquaculture to more exposed locations. Since water currents are generally stronger out in the open sea when compared to the traditional fjord sites, there are concerns of reduced growth, injuries, or even mortality caused by swimming fatigue in high currents over long periods of time (Bjelland et al., 2015).

Several fish-farming localities have water current velocity profilers placed near the site, running continuously. Therefore, measurements of water current strengths and their direction are in many cases available, but only a few products on the market measure the swimming speed of individual Atlantic salmon. During a conversation with Daniel Engen Lauritzen, a fish-health biologist working at Sinkaberg Hansen AS, he informed that a system that can show the swimming speed of individuals over time would be beneficial to their assessment of welfare. To paraphrase: "*Such systems could yield swimming speed trends over time for fish in the same pen. Since the physical strength, aquatic parameter tolerance, and sometimes general behavior of Atlantic salmon from different genetic breeds vary, such trends are in many cases more valuable than absolute single-point measurements on a selection of individuals.*"

Although $\mathrm{U_{crit}}$ is a measure that is mostly used for characterizing maximum water current tolerances, the general swimming activity dynamics of Atlantic salmon have been researched to evaluate its connection to stress. E. Svendsen et al. (2021) performed controlled environment tests to measure the heart rate and swimming activity of Atlantic salmon undergoing stressing events. The Atlantic salmon were placed in water tanks and implanted with accelerometer and heart rate tags to measure these two parameters. To induce stress, the water tank levels were reduced until the dorsal fins were exposed to air. The heart rate showed a significant increase during the stress-events, and a delayed response in increased swimming activity also occurred. It was still concluded that the increased swimming activity was a consequence of stress.

Crowding is the procedure of using a separate net in the fish pen to pull the fish towards the extraction pump. This process speeds up the collection of fish for slaughter or de-lousing operations but is in many cases a cause of substantial losses of salmon due to the side effects of heavily increasing clustering density. Both stress from clustering and asphyxiation due to low O_2 availability are direct causes of death in these situations. A full-scale experiment during a crowding and de-lousing operation with tagged fish was conducted by Føre, Eirik Svendsen, et al. (2018), where the swimming activity and heart rate of 21 fish were logged during three separate such operations. The swimming activity was significantly higher during the crowding and de-lousing operations when compared to the background levels measured one day prior, indicating that swimming activity is linked to stress.

The relationship between Atlantic salmon welfare and swimming activity is far from trivial. Additionally, the relationship might not be causal in all situations and additional data could be necessary to draw conclusions on welfare. Nevertheless, if the physical behavior of a group of animals is measured and logged, changes in behavior over time can give objectively observable trends which could be a better indicator of deteriorating, stationary, or improving welfare.

2.5.2 Vital Aquatic Parameters

Oxygen is vital for living organisms to survive. Dissolved Oxygen (DO), which is the amount of free O_2 molecules in the water, is a key factor that affects the welfare and growth rate of Atlantic Salmon (Burt et al., 2012; Oppedal et al., 2011). DO is either represented as an O_2 saturation percentage or as a true measure, such as milligrams of O_2 per liter of water (mg/L). Full O_2 saturation (100 %) means that the water is holding as many O_2 molecules as it can in equilibrium, but since this is a chemical process dependent on salinity, temperature, and pressure (Xing et al., 2014), 100 % O_2 saturation in two different locations at sea does not mean the water contains the same amount of free O_2 molecules. Furthermore, the amount of DO in the ocean, and therefore within aquaculture sea pens, fluctuates and is dependent on other factors such as the local weather, sunlight, ocean currents, and oxygen demand from living organisms in the ocean (Johansson et al., 2006).

In literature, there seem to be several different conclusions as to what the optimal O_2 saturation for Atlantic salmon is. Generally, most cited papers show that an O_2 saturation > 75 % is considered satisfactory, given that other aquatic parameters are also within their acceptable ranges. Oppedal et al. (2011) reviewed several behavioral traits and environmental parameter thresholds for Atlantic salmon, where it is cited that a DO saturation of 70 % resulted in reduced appetite and at 60 % acute anaerobic metabolism occurred. This experiment was however conducted in seawater at a temperature of 16 °C, which emphasizes the importance of the quality of other aquatic parameters in addition to DO. Burt et al. (2012) studied the DO levels in aquaculture sea pens in Newfoundland, where several occurrences of hypoxic conditions were recorded. In this paper, the hypoxic conditions were defined as DO $<$ 55 %, 61 %, 66 %, 72 %, 78 % at 2, 6, 10, 14 and 18 °C, respectively. Numerous other research papers have found similar results where hypoxic conditions have occurred in operational fish pens.

Due to a rapid increase in lice infestations over the past years (Føre, Frank, et al., 2018), fish farmers have started to use specially developed skirts surrounding the fish pen. These skirts have been reported by the industry to decrease the lice-infestations, but experiments conducted by Stien et al. (2012) revealed that the O_2 saturation decreased from 88.3 % to 64.2 % after one day of deployment of the skirt. Results of the same magnitude were found by Jónsdóttir et al. (2021), where the DO saturation increased from 59 % to 81 % within 30 minutes of the skirt being removed.

Extreme hypoxic events could in the worst case lead to mortality and hypoxia in general stresses the fish. Non-lethal hypoxia results in behavioral changes in all cited work related to the subject, where reduction in appetite and activity, as well as escape attempts, have been recorded (Johansson et al., 2006; Hvas et al., 2019). Therefore, monitoring O_2 levels in aquaculture sea-pens is essential for assessing Atlantic salmon health and welfare, but also for optimizing the feeding process and hence growth rate due to the effects low DO levels have on appetite.

Oppedal et al. (2011) cites that water temperature can be the decisive environmental parameter that affects the salmon swimming depth and density preferences. The water temperature in the water column within fish pens varies hourly, daily, and seasonally. Falconer et al. (2020) details how temperature variations heavily affect the health and welfare of Atlantic salmon. This study focuses on how future climate change can affect the aquaculture industry, where Table 2.1 shows how water temperature is a vital parameter for health and welfare.

Temperature	**Effects on salmon**
>20°	Growth stops, mortality
16°-20°	Reduced welfare and food intake, slow growth, increased stress and mortality
14°-16°	Sub-optimal growth, risk of reduced health and welfare
11°-14°	Optimal growth and food intake
7°-11°	Sub-optimal growth, risk of reduced health and welfare
<7°	Reduced welfare and food intake, slow growth, increased stress and mortality

Table 2.1: Temperature thresholds for Atlantic salmon, adapted from Falconer et al. (2020).

From all cited work presented above, it is arguably clear that O_2 and temperature monitoring is very important in fish farms which, as previously mentioned, are responsible for millions of individual fish simultaneously.

2.6 Deep Learning and Computer Vision

This section will first briefly present the field of Computer Vision and Deep learning before introducing artificial and convolutional neural networks. Then, visual object detection and the You Only Look Once visual detection framework is detailed, with emphasis on its applications in this project. Lastly, visual multiple object tracking is presented. Due to the practical approach in the project and utilization of open-sourced software, the presented theoretical background is simplified to a level that yields a basic understanding of how the Computer Vision and Deep learning pipeline presented in Chapter 3 can be re-created and applied by the reader. For a deeper dive into the world of Computer Vision, Deep learning, and their applications, the reader is recommended to take a look at the free downloadable book published by Szeliski (2011).

Computer Vision (CV) is a field of Artificial Intelligence (AI) that focuses on information-extraction on digital images or video, where the computer in essence can reproduce some of the abilities of human sight. These abilities are most commonly detection and classification of objects, where the computer can understand characteristics of the visual environment. The CV information extraction has in recent years often been performed by applying deep learning methods. Deep learning is a term describing the application of ”deep networks”, which take architectural inspiration from how the brain works. These networks mimic how brains process information by using what are called artificial neural networks. The specific terminologies for these different, but still conceptually similar fields, have become increasingly overlapping with the advancements they have undergone since they were initially termed.

The networks used in Deep learning are often composed of many successive layers between the input and output, hence the term *Deep* learning. Deep learning and CV in industrial and commercial settings have seen a large increase over the past years, where the applications range from self-driving assistance in commercial vehicles to classification of wild animals (Goodfellow et al., 2016, pp. 455-456). Most Deep learning-based CV object detectors follow a similar routine in how they are constructed, where the following four steps generalize the process:

1. Gather data for training (images).

2. Label the data as it should look in the end result.

3. Train a neural network with the labeled data until it achieves desirable results.

4. Use the trained network to detect and classify objects in un-labeled images.

The detection and classification accuracy of Deep learning networks, in general, relies heavily on the model itself and the training routine, where there are thousands of different network architectures and design methods which achieve various results. To gain a bit of a general understanding of how Deep learning works before CV-specific networks are detailed, we start with a generic artificial neural network introduction.

2.6.1 Artificial Neural Networks

A traditional Artificial Neural Network (ANN) consists of several layers each containing a set amount of neurons. The outputs of all neurons within each layer are connected to the inputs of the neurons in the next layer. Specifically, this is a fully connected feedforward ANN, also called a ”multi-layer perceptron”. Only the input (first) and output (last) layers are visible, while all layers in between are hidden (Goodfellow et al., 2016, pp. 167-168; Szeliski, 2011, pp. 271-272). This architecture is illustrated in Figure 2.11.

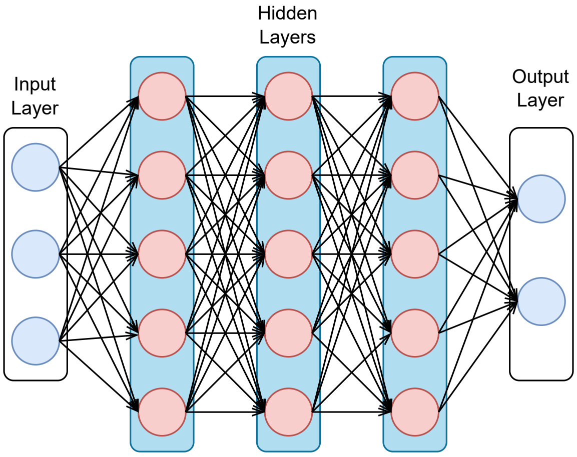


Figure 2.11: Illustration of a fully connected neural network with 3 hidden layers.

The neurons in each ANN layer perform a weighted sum of their input values, and this result is passed through an activation function. The activation function output is then fed forward to the next connected neuron until all information has reached the output layer. One input to output information transmission is called a feedforward pass. The individual neuron weights are what determine how the neural network classifies objects (Goodfellow et al., 2016, p. 270). The neuron weights within each layer can be organized as a vector, such that the computation performed in each layer, *l*, is a matrix multiplication, i.e. (Goodfellow et al., 2016, p. 271):

$$s_l = \mathbf{W}_l \mathbf{x}_l \tag{2.12}$$

where
$\mathbf{W}_l$ = Weight matrix for layer l.
$\mathbf{x}_l$ = Inputs to layer l.
s_l = Weighted sum for layer l.

The hidden layers within the network then use the inputs to compute the output based on the individual neuron weights and activation functions.

Training neural networks mainly consist of feeding the network with labeled data and iteratively tuning the neuron weights until the output converges to the desired accuracy. As an example, the input to the illustrated ANN could be weight, length, and color, called features, and the possible outputs are salmon or cod, called classes. To train this network to discriminate the two classes, the network should be given a large set of these inputs with their corresponding correct class and iteratively adapt the neuron weights until it gives the desired accuracy on correct classification. The computational goal during training is to minimize a loss function, which is a numerical value associated with how ”wrong” the network weights currently are. Since the network does not know which class the labeled data belongs to during the feedforward process, the loss value is computed at the end of every feedforward pass by comparing the output result in each pass with the actual class label. The neuron weight-tuning itself is then performed by utilizing *gradient descent* and *backpropagation.* Gradient descent is used to minimize the loss function, where the derivative of the loss function for the current data sample with respect to the network weights is computed. This derivative is then propagated back through the network to alter the weights to improve the output accuracy, hence *backpropagation.* When the desired classification accuracy is achieved, the network is successfully trained with a set of weights that should be able to classify new and unlabeled inputs correctly (Szeliski, 2011, pp. 279-289).

2.6.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a type of ANN which are specialized to process images or grid-shaped data in general. Compared to traditional ANNs which perform matrix multiplications with the neurons in its layers, as shown in Equation 2.12, CNNs use the convolution operation instead. The layer structuring in CNNs also differs from ANNs, where each layer is organized as feature maps. The feature maps are generated by iteratively performing a convolution between a small area of the image and a *kernel*, working its way across the entire image. This generates overlapping local feature maps which are allocated to the respective spatial locations in the image, such that the combination of these local features can produce discriminative features, such as edges of an object (Goodfellow et al., 2016, pp. 331 - 340; Szeliski, 2011, pp. 290 - 294).

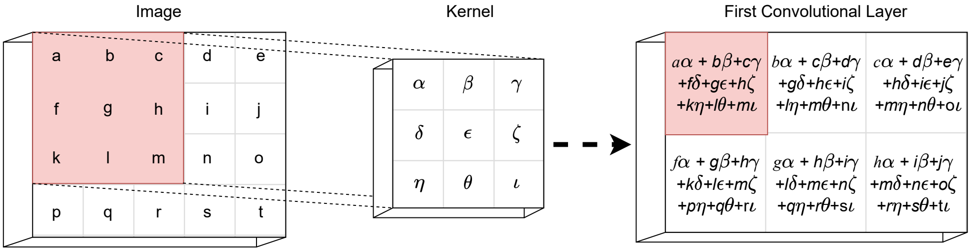


Image
a b c d e
f g h i j
k l m n o
p q r s t
Kernel
α β γ
δ ϵ ζ
η θ ι
First Convolutional Layer
aα + bβ+cγ +fδ+gϵ+hζ +kη+lθ+mι
bα + cβ+dγ +gδ+hϵ+iζ +lη+mθ+nι
cα + dβ+eγ +hδ+iϵ+jζ +mη+nθ+oι
fα + gβ+hγ +kδ+lϵ+mζ +pη+qθ+rι
gα + hβ+iγ +lδ+mϵ+nζ +qη+rθ+sι
hα + iβ+jγ +mδ+nϵ+oζ +rη+sθ+tι

Figure 2.12: Illustration of CNN kernel convolution.

Most CNNs include multiple convolutional layers, where each layer has a set amount of kernels with a fixed size. Typically, kernels with size 3x3 are used in the feature map convolutional layers while 1x1 kernels are used to downsample the feature maps. The 1x1 convolutions maintain the feature map size, but rather reduce the feature map depth to decrease computational complexity (Szegedy et al., 2014; Szeliski, 2011, pp. 300-301). Additionally, there are often *pooling layers* between (or just after) the convolutional layers. The pooling layers essentially alter the feature map size by extracting one value from each feature map in the current layer and passing it to the next convolutional layer. The most common pooling layer function is named *max pooling*, where the largest value in each feature map is extracted and passed on to the next convolutional layer. This results in a local invariance to translation, such that a small translation of the input still yields the same pooled output at the spatial location in the image. Furthermore, pooling is important to allow the network to process images of varying size, since the final classification layer of CNNs have a predetermined size (Goodfellow et al., 2016, p. 343).

Gradient-based training of deep neural networks, as presented in the previous section, can lead to an accuracy saturation where the loss minimization converges poorly or even becomes unstable. This is caused by the *exploding/vanishing gradient problem*. During backpropagation, the gradient is used to update the weights in all layers. Since each layer multiplies its weights with the input value (gradient in this case) before passing it on to the next layer, a long line of small or large successive multiplications will cause the updated weight values in the deeper layers to become vanishingly small or very large. The former causes the loss minimization to stall, while the latter makes the learning unstable (Goodfellow et al., 2016, p. 290; He et al., 2015).

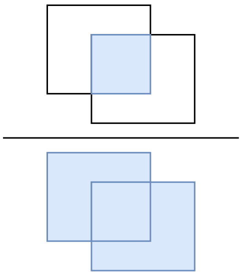
He et al. (2015) presented deep residual learning as a method to alleviate deeper networks from such issues during gradient-based training. In residual networks, *skip connections* allow information and gradients to skip certain layers, such that the network layers learn to be influenced from a layer N steps above/below (Goodfellow et al., 2016, p. 410; Szeliski, 2011, pp. 301-302; He et al., 2015). Many deep neural network architectures utilize these skip connections to alleviate the aforementioned issues.

2.6.3 Object Detection in Computer Vision

Up until this point, the presented theory has been purely related to what is called the *backbone* in a Deep learning framework. These fundamentals are essentially what occurs at the core of a CV Deep learning pipeline. There are however a large variety of CNN purposes within CV and Deep learning. Network architectures and depth, as well as CV detectors, may be very different but still designed to achieve the same goal. This report mainly focuses on object detection and tracking. Thus, the following theory only discusses such applications.

In the context of visual object detection and classification, the main goal is to pass an image containing different objects into the network and get an image out that has bounding boxes which encapsulate the detected objects with their class label. The most commonly used performance metrics associated with object detectors are their inference speed, i.e. the amount of time it takes to compute the result, and their accuracy, given by the mean average precision (mAP) in a given dataset. The mAP is computed by measuring the precision of the detector at different Intersection over Union (IoU) thresholds for all classes in the dataset. The IoU is a metric that describes how accurate the bounding boxes are, where the goal is to have a box that perfectly encapsulates the outer edges of the object (Szeliski, 2011, pp. 370 - 380). The IoU is computed by comparing the predicted bounding box with the manually labeled bounding box during training or testing, also named the *ground truth*, i.e. (Szeliski, 2011, p. 380):

$$\text{IoU} = \frac{Overlapping\ Area}{Area\ of\ Union} = \tag{2.13}$$



If the IoU value for a prediction is higher than the current IoU threshold and the class label is correct, the prediction is assigned as a True Positive (TP). If the IoU value is lower than the threshold, the prediction is assigned as a False positive (FP). Assigning P to be the number of labeled detections in the current image, the *precision* and *recall* can be computed for each image, i.e. (Szeliski, 2011, pp. 379-380):

$$\begin{aligned} \text{precision} &= \frac{TP}{TP + FP} \\ \text{recall} &= \frac{TP}{P} \end{aligned} \tag{2.14}$$

Populating a precision-recall curve for these calculations at different thresholds and computing the area under the curve yields the Average Precision (AP). Repeating this for all the different classes and taking the mean results in the mAP (Szeliski, 2011, pp. 380-382).

Two large datasets which are commonly used as a benchmark on different CV object detectors are the PASCAL Visual Object Classes (VOC) Dataset (Everingham et al., 2010) and the Microsoft Common Objects in Context (COCO) Dataset (Lin et al., 2014) (Szeliski, 2011, pp. 379-380). Figure 2.13 shows a graphical leaderboard of the historical and current top performers on the COCO dataset, from January 2018 to April 2022.

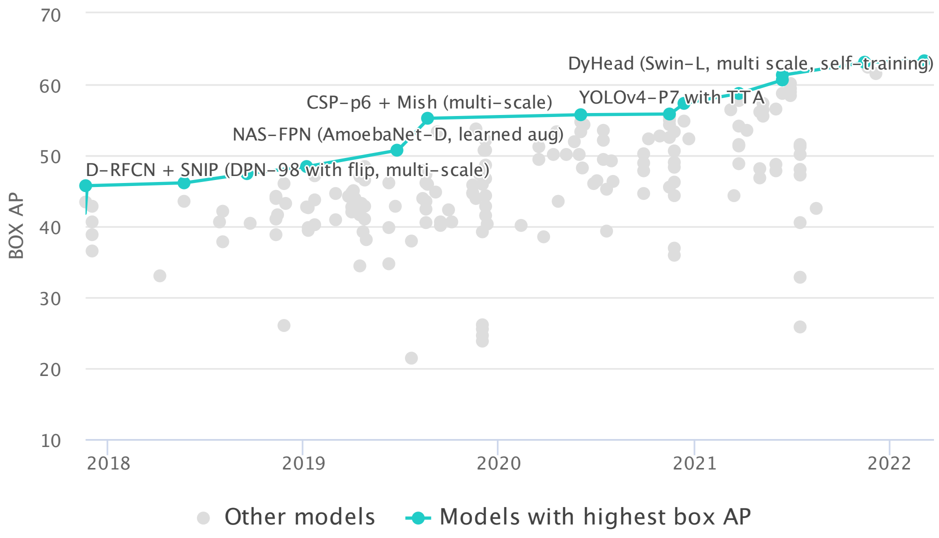


Figure 2.13: COCO Object Detector Leaderboards, from *Object Detection on COCO test-dev* (2022).

There are two main categories of visual object detectors: region-based and regression-based. The region-based methods are also called two-stage detectors since they perform detection and classification in two separate stages. First, a network generates proposed bounding boxes containing objects within the image and then passes these regions in to a separate classifier network to assign class labels to the objects. Regression-based detectors instead generate both bounding boxes and class labels in one unified pipeline and are therefore also called single-shot detectors (Szeliski, 2011, pp. 382-386). Consequently, single-shot detectors have higher inference speeds and are preferable if computational efficiency is equally or more important than maximizing accuracy.

2.6.4 You Only Look Once

You Only Look Once (YOLO) is a single-shot real-time object detector that has seen large public interest since its initial publication. The first version of YOLO was published in 2015, which at that time used a network with 24 convolutional layers and max pooling between each layer. The performance of YOLOv1 was compared to other state-of-the-art detectors, where promising results were achieved. When compared to other real-time detectors at the time, both the precision and inference speed of YOLOv1 was significantly higher. It achieved a mAP of 63.4 at 45 frames per second (FPS), whereas the nearest (cited) competitor network, DPM, achieved a mAP of 26.1 at 30 FPS in the PASCAL VOC dataset. The best region-based detector at the time, Faster R-CNN VGG-16, achieved a mAP of 73.2 at 7 FPS (Redmon, Divvala, et al., 2015).

Over the course of the next years, many advancements were made in the field of Deep learning, and a variety of new real-time detectors were published and released into the open-source domain. A second (Redmon and Farhadi, 2016) and third (Redmon and Farhadi, 2018) version of the YOLO detector were published, where its performance in both releases was on par or better than other current state-of-the-art detectors. YOLOv3 was the last version of YOLO which was published by the original author. This version presented several improvements, pushing the YOLOv3 performance beyond all (cited) competitors on inference speed while maintaining a comparable or higher mAP when tested on the COCO dataset (57.0 mAP at 51ms inference time) (Redmon and Farhadi, 2018). YOLOv3 also introduced a new network architecture; Darknet-53.

Darknet is a ”*neural network framework written in C and CUDA*” (Redmon, 2013–2016). CUDA (Compute Unified Device Architecture) is NVIDIA’s Graphical Processing Unit (GPU) Application Programming Interface (API). GPUs handle image processing-related tasks better than CPUs because they are tailored for images by design, and computation time is therefore heavily reduced if GPU acceleration is used. It is therefore ideal to run image-targeted neural networks with inherent CUDA support if an NVIDIA-based GPU is used.

Darknet-53 has 53 convolutional layers and takes architectural inspiration from the YOLOv1 and Darknet-19 network introduced in YOLOv2 (see (Redmon and Farhadi, 2016)). The Darknet-53 network abandoned max pooling and instead includes skip connections through residual blocks, as presented in Section 2.6.2. Furthermore, Darknet-53 makes object detections at three different image scales to improve the detection of small objects (Redmon and Farhadi, 2018).

Darknet-53 is open-sourced[6], such that anyone can compile it on their computer and run YOLOv3 with Darknet-53 as the backbone with a pre-trained model to run detections on arbitrary images. Additionally, a full guide on custom training is given such that the network can be trained to detect custom objects.

The final official version of YOLO, YOLOv4, introduces a significant increase in mAP and minor inference speed improvements. A YOLOv4 performance benchmark comparison on the COCO dataset is shown in Figure 2.14. In this version, the network is split into three different parts, where a modified version of Darknet-53, CSPDarknet53 (Wang et al., 2019), is used as the backbone. The YOLOv4 network architecture is quite extensive compared to the previous versions, and the reader is therefore referenced to Bochkovskiy et al. (2020) for the complete architecture presentation. YOLOv4 is open-sourced[7] similarly to YOLOv3, and implementations in a variety of frameworks, such as OpenCV and TensorFlow for Python are included.

[6] `https://github.com/pjreddie/darknet`

[7] `https://github.com/AlexeyAB/darknet`

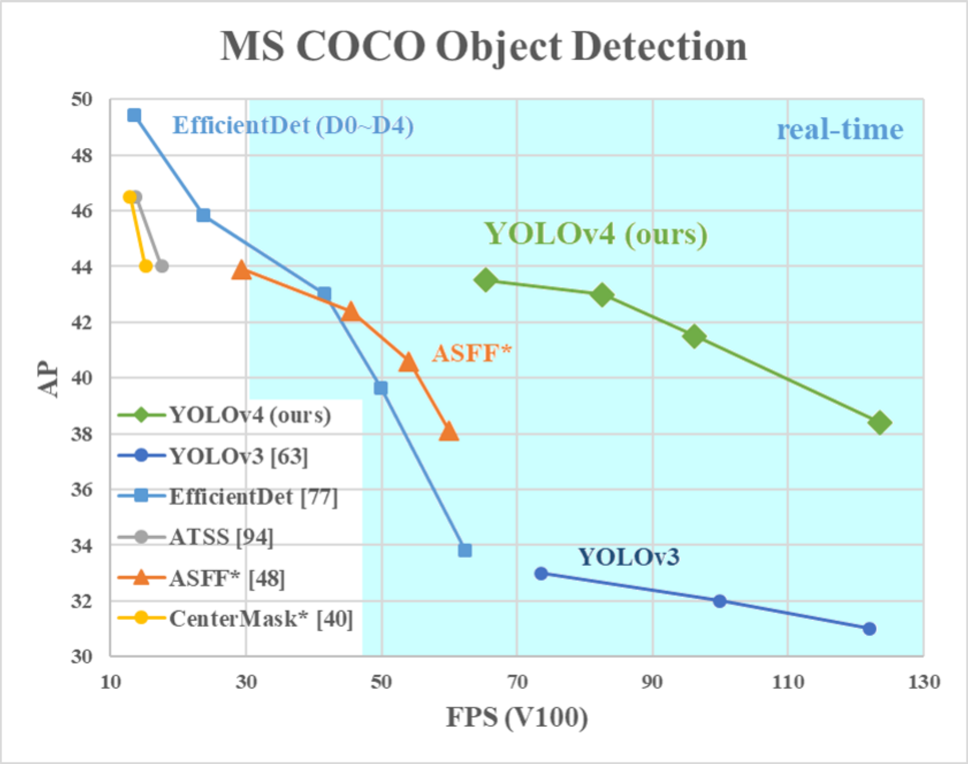


Figure 2.14: Performance of YOLOv4 vs. other object detectors, from Bochkovskiy et al. (2020).

2.6.4.1 Training of Custom Object Detectors

Similar to the example mentioned in Section 2.12 about training an artificial neural network, CNN training is in essence quite similar. Instead of feeding the network the weight, length, and color measures of the two fish species, it is given images as input with pre-labeled bounding boxes around the different fish species with an associated class label (salmon or cod). The main difference in the backbone is that the neuron weights are now the kernel values in all layers, which are tuned to minimize the loss during training. As mentioned in the preceding section, the Darknet-53 architecture (and CNNs in general) include several layers with numerous kernels, such that all kernels in the same layer are trained in parallel. This results in the network learning many ways of interpreting the images, where the different kernels learn different specific characteristics of the objects.

Training object detectors on custom datasets where the network does not know how to detect the new class of object is very often done through a process called *transfer learning*. Transfer learning is the procedure of using a pre-trained model to learn the characteristics of new and unknown objects. The ideology behind this fact is that the pre-trained model already knows how to detect certain objects, and it is desired to adapt it to new objects. Since objects, in general, can have similar ”low-level characteristics” such as edges, shapes, geometry, and changes in lighting (Goodfellow et al., 2016, p. 539), the model can adapt and modify its weights to learn the similar features that the new and previously unknown objects exhibit (Goodfellow et al., 2016, p. 539; Szeliski, 2011, p. 313). This also saves computation time during training since the network does not have to be trained from scratch.

2.6.5 Multiple Object Tracking

Multiple Object Tracking (MOT) in videos is a challenge that requires assigning information about detections in the current frame with other video frames. There are two different methods for object tracking; the batch method and the online method. The batch method uses future information to yield estimates on the current frame, while the online method uses the present frame and past information to estimate future information. Batch methods are intuitively better, but future information is not available in systems that are designated to operate in real-time (Chen et al., 2019). Therefore, only the online method is presented.

As mentioned above, online trackers use information from the current and past frames to predict information in the next frame. This methodology requires a pre-determined state model of the targets to track. The Simple Online and Realtime Tracking (SORT) algorithm, presented by Bewley et al. (2016), was specifically developed for MOT in CV applications. By using a constant velocity state model of the targets, a Kalman filtering approach is used to track objects.

The Kalman filter is a very popular algorithm to apply in state estimation for tracking objects (and more). They operate under the assumption that the state dynamics are discrete-time linear dynamical and that the measurement noise is random, independent Gaussian (Jaulin, 2015, p. 232). Hence, this constant speed assumption is applied to the movement dynamics of the Atlantic salmon in this thesis. The Kalman filter is a recursive algorithm that performs two distinct state-operations in every iteration; *prediction* and *correction*. Assuming that current state information for a tracked object is available ($\mathbf{s}_{\mathrm{t=0}}$), the *prediction* phase uses this state and the measurement noise model to predict the next state ($\mathbf{s}_{\mathrm{t=+1}}$). The *correction* phase incorporates sensor measurements to more accurately update the state. If no measurement of the objects position is available at t=+1, the next state (now $\mathbf{s}_{\mathrm{t=+2}}$) is predicted purely based on the state before. If a measurement of the objects position is available at t=+1 however, this sensor measurement is used to *correct* the t=+1 state and noise model to improve the t=+2 state prediction (Kalman, 1960). The above description of the Kalman filters is arguably excessively simplified, but illustrates the basic ideology of how they work to understand its relevance in the work presented later in the thesis. The first step of utilizing Kalman filtering for tracking is to create a state model describing the dynamics of the objects to be tracked.

The state model of each target in the SORT algorithm is defined as (Bewley et al., 2016):

$$\mathbf{x_N} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{2.15}$$

where
$\mathbf{x_N}$ = State of target N.
u = Target bounding box center position (horizontal).
v = Target bounding box center position (vertical).
s = Target bounding box scale.
r = Target bounding box aspect ratio.
$\cdot$ = Velocities of the respective parameters.

Assignment of a unique ID to a new individual occurs based on an IoU threshold (see Section 2.6.3). If a detection in the current frame doesn't have an IoU overlap with any other detections which is larger than the threshold, it is assigned a new ID. The assignment of the same ID to detections in two subsequent video frames is based on the IoU distance between the prediction of each target's bounding box location and all input detections in the current frame. Hence, a target is tracked with the same ID if the predicted state parameters closely match the parameters of a detection in the current frame from the underlying object detector. Tracked targets are terminated if they are lost in one frame, such that very short occlusions will cause the object to receive a new ID after it has reappeared. In the SORT paper, the authors inform that *"object re-identification is beyond the scope of this work."* (Bewley et al., 2016).

Re-identification of tracked targets after temporary occlusion is highly relevant for MOT with many targets at different distances from the camera in e.g. pedestrian or fish tracking. This shortcoming of the SORT algorithm was addressed by Wojke et al. (2017), where a supplementary appearance descriptor is added to the state model. This algorithm is named *DeepSORT*, since it uses the SORT algorithm with a *deep association metric*. The motion-based state space is identical to the one used in SORT (Equation 2.15). Additionally, a CNN is used to compute appearance descriptors for all bounding boxes in parallel with the motion-based tracker, where the last 100 appearance descriptors for each track are stored. The assignment of IDs to tracked targets is then given through a weighted sum combination of the two metrics. The motion-based assignment is identical to SORT and the appearance-based metric complements ID assignment through the appearance-similarities between two frames and targets through a similarity-comparison (smallest cosine distance). The motion-based information is effective in frame-to-frame tracking without occlusions, and the appearance-based information is effective at recovering IDs of objects that have been occluded (Wojke et al., 2017). Table 2.2 shows the performance of a few batch-based and online trackers tested on the MOT16 challenge. The performance metrics used are Accuracy (MOTA), Precision (MOTP), Number of ID switches

(ID), and average FPS (Runtime) (Milan et al., 2016).

	Method	MOTA	MOTP	ID	Runtime
LMP_p	**BATCH**	71	80.2	434	0.5 Hz
NOMTwSDP16	**BATCH**	62.2	79.6	406	3 Hz
POI	**ONLINE**	66.1	79.5	805	10 Hz
SORT	**ONLINE**	59.8	79.6	1423	60 Hz
DeepSORT	**ONLINE**	61.4	79.1	781	40 Hz

Table 2.2: Tracking results on MOT16 Challenge. Adapted and modified from Wojke et al. (2017).

Since the DeepSORT state model parameters are accessible outputs from the YOLO detection framework (bounding box parameters), and the computation speed is high both in YOLO and DeepSORT, they can be layered without heavily increasing the total processing time.

This concludes all the theoretical aspects which will be implemented later in the report. However, before the practical applications are presented, the hardware it is applied on is detailed.

Chapter 3

The FISC System

This chapter presents the design and functionality of the Fish-farm Integrated Sensor Cluster (FISC) prototype. The hardware and acquisition software design of the FISC system has been detailed in the preceding specialization project report (see (Ericsson, 2021)), and has barely been altered since then. A simplified summary of the system is presented to begin with, such that the reader can understand the functionality and performance of the system from this report alone. The theoretical aspects behind the design choices in the previous work are not detailed but can be seen in the aforementioned specialization project report. To differentiate previous work and new thesis-specific implementations, the presentation below is simply split into different sections. Lastly, the details surrounding two performed full-scale field tests are presented.

3.1 Previous Work

The FISC system prototype consists of three separate parts; 1) the FISC capsule, 2) an interface cable and 3) the topside cabinet.

3.1.1 The FISC Capsule

The FISC capsule is the housing containing all but one of the sensor components in the system. The housing is made of Aluminium Bronze, a material that is resistant to corrosion in seawater. It is deployed in the water column by suspending it from the interface cable within the fish pen to gather aquatic environmental data, orientation-information as well as direct data from fish (optically and acoustically).

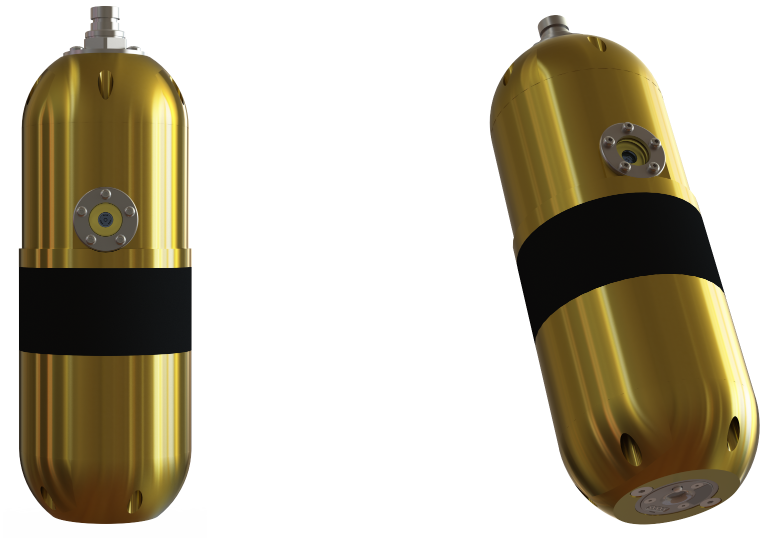
The FISC capsule includes the following sensors and modules:

- 8 directional horizontal echosounder receivers, hereby named *Sectors*.
- A low-cost camera centered above Sector #4 (Raspberry Pi Camera Module V2).
- A Raspberry Pi 4 Model B (main capsule processor).
- An O_2 and Temperature-sensor from Aanderaa (Oxygen Optode 5730).
- A custom PCB including an MCU, a 9DOF orientation sensor (BNO055 from Bosch Sensortec), an analog frontend for the acoustics (Variable Gain Amplifier (VGA)), and additional communication and multiplexing interfaces.

The design, electrical specifications, and implementation procedure of the custom PCB with its components, as well as the acoustic design are detailed in the specialization project report (Ericsson, 2021). The echosounder receivers are multiplexed, where two and two sectors are sampled simultaneously. Consequently, it takes four transmissions to complete a full 360° acoustic scan. The Raspberry Pi functions as an interface between the topside cabinet and the PCB, where a USB interface between the Raspberry Pi and the PCB MCU controls and configures the MCU software. Additionally, the Raspberry Pi directly samples the O_2/Temperature-sensor and camera module and passes this data through to the topside cabinet. The communication between the FISC capsule and the topside cabinet is through an Ethernet interface, where the Raspberry Pi is connected to the PC in the topside cabinet through internal Cat5 wiring in the interface cable. The custom network interface and software commands are also presented in the aforementioned specialization project. A simplified block diagram of the FISC system can be seen in Appendix A.

Figure 3.1 shows 3D renders of the FISC capsule, where the camera lens and O_2/Temperature-sensor are seen in the center and on the bottom of the capsule, respectively.

The sensitivity calibration and horizontal beam angle characterization of the acoustic receivers were performed in the specialization project and can be seen in Appendix B and Appendix C, respectively. The most important fact to take from these calibration results are that the peak sensitivity occurs at 470 kHz and that the Sectors have approximately 5-6 dB acoustic overlap at the same frequency. The former is taken into consideration in determining which center frequency is used for acoustic transmission, and the latter shows that the radial acoustic coverage is 360 °, which was intended by design.



(a) Front view. **(b)** Diagonal view.

Figure 3.1: 3D Renders of FISC capsule.

3.1.2 Topside Cabinet and Interface Cable

The topside cabinet contains the following modules:

- A SuperServer E100-9W-H PC (topside PC).
- A 4G/Wifi Access Point (AP) for internet connectivity.
- Main power supply for topside PC and AP (230V to 24V/5V).
- Secondary power supply for FISC capsule (230V to 12V).
- An Analog Discovery 2 USB oscilloscope.
- A transmission power amplifier (TXPA) with output transformer for acoustic transmitter (TX).

The topside PC is the main controlling unit in the system. It mostly runs the custom Python acquisition scripts (detailed in the next section), where the 4G AP makes it possible to control remotely through a Secure Shell or TeamViewer connection. The Analog Discovery 2 (AD2) is responsible for generating the TX pulse and trigger signal to the capsule custom PCB, as well as sampling the two and two multiplexed receiver sectors (analog voltage). It is controlled through a Python interface through the included AD2 Software Development Kit (SDK). The TXPA with output transformer amplifies the transmission pulse amplitude from 3.3 V to approximately

160 V, achieving a propagated acoustic signal sufficient for detecting small targets. (For a more detailed description, see the presentation in (Ericsson, 2021)).

The FISC interface cable design is based on cables that are used for several of Norbit's existing products, but with a custom pinout on both ends. Additionally, the acoustic transmitter (TX) is integrated to the cable itself, where it is molded 10 cm above the bulkhead connection to the capsule. The bulkhead connections used in the interface cable and their termination path are as follows:

- 2x Cat5 Twisted Pairs for Ethernet link: Topside PC to capsule Raspberry Pi.
- 2x Shielded Twisted Pairs for analog acoustic Receiver (RX) signals: Topside AD2 to capsule custom PCB.
- 3x Shielded Twisted Pairs for analog acoustic transmitter signal (TX): Topside TXPA to integrated TX on cable.
- 1x Shielded trigger wire: Topside USB oscilloscope to custom PCB.
- 2x Power Lines (12V + GND): Topside 12V power supply to the FISC capsule voltage regulator.

The first version of the FISC interface cable, which was constructed and presented by Ericsson (2021), is not used anymore. The internal pinout listed above relates to the new interface cable version (Rev. 2), which was constructed in February 2022. The reason for the construction of a new cable was mainly due to issues with electromagnetic interference within the cable caused by the high voltage acoustic transmission pulse. This interference was substantial enough to break the Ethernet link between the topside PC and capsule Raspberry Pi, which essentially froze the acquisition software until the link was reestablished (30-60 s).

Although this case probably should have been presented as new work with a more detailed root cause analysis, it is instead simply introduced here since it does not add any substance to the main goals of the thesis. Figure 3.2 illustrates the changes performed to combat this issue, which ultimately worked as intended. By changing the pinout such that the transmitter leads have grounded shielding, most of the electromagnetic interference takes this path to ground rather than affecting the other signal wires within the cable. Three separate 26 AWG pairs were used to minimize the total impedance in the TX wires.

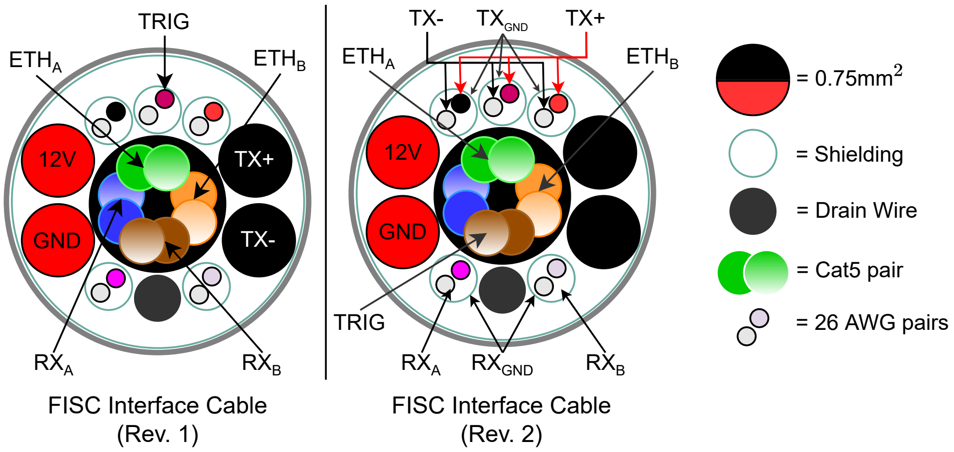


Figure 3.2: FISC interface cable revision differences.

Figure 3.3 shows an image of the FISC capsule with the interface cable connected. The TX is visible above the bulkhead connector. The gaffer tape on the receivers are modifications to the acoustics which will be elaborated in Section 4.1. The acoustic characterization of the transmitter was originally performed in the specialization project but had to be repeated with the new interface cable since the transmitter ceramics were reconstructed and molded on the new cable with an identical procedure. The resulting transmitting voltage response and directivity of the Rev.2 TX can be seen in Appendix D and Appendix E, respectively.

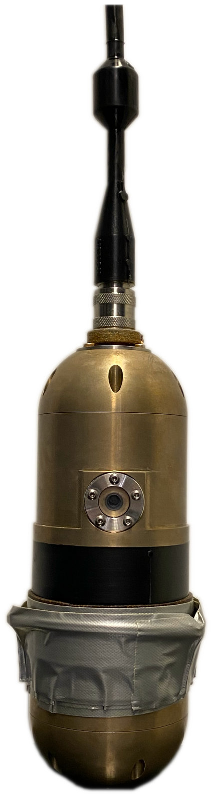


Figure 3.3: Image of FISC capsule with interface cable.

3.1.3 Acquisition Software

The Python software which controls the data acquisition and system configuration has barely been altered since the presentation in Ericsson (2021). The bulk of work on this thesis is related to the post-processing of acquired data.

The latest version of the data-acquisition software runs by swapping between two different modes every N seconds. These modes are:

1. ***Sector Scan***
   - Record video (for N seconds, user configurable).
   - Sequentially transmit acoustic signal and sample two-and-two RX Sectors.
   - Sample orientation sensor between every acoustic ping.
   - When N seconds have passed, sample O_2/Temperature-sensor and pass this data to topside PC through Ethernet. Store data to topside PC hard drive and swap mode.
2. ***Sector Focus***
   - Record video (for N seconds, user-configurable).
   - Repeatedly transmit acoustic signal and only sample the RX sector which is below the camera.
   - When N seconds have passed, sample orientation and O_2/Temperature-sensor and pass this data to topside PC through Ethernet. Store data to topside PC hard drive and swap mode.

A flow-chart of the acquisition software pipeline can be seen in Appendix F. After every Sector Scan and Sector Focus routine is completed, a compressed file is saved to the topside PC with a timestamp. Each file contains the data shown in Table 3.1. These generated files, as well as the recorded videos, are what the post-processing software takes as input.

Header Content	**Raw Acoustic Data**
- TX pulse parameters. - Center frequency, BW and length. - AD2 sampling rate. - Sound speed in water (custom value). - Acoustic sampling range. - Unit Quaternion sample. - O_2 and temperature sample.	- 2D if Sector Scan (8 columns, all Sectors). - 1D if Sector Focus (Sector 4 only).

Table 3.1: FISC acquisition file contents.

3.2 Software Implementation and Post-processing

As previously mentioned, the bulk of this thesis work relates to the post-processing of acquired data. This section presents which frameworks are used and how they are implemented in the FISC system to achieve the results presented in Chapter 4. It is important to note that this project in its entirety, i.e. everything from the hardware design and post-processing, is designated as a proof of concept for a future product on the market. The desire to achieve results that can be directly applicable for future work in product development outweighed the traditional ideology of constructing everything from the bottom up in thesis-related work. Therefore, open-sourced software libraries and GitHub repositories have been used when possible, where necessary modifications to tailor the software for the desired functionality have been performed.

3.2.1 Custom YOLOv4 Object Detector Implementation

The complete YOLOv4 framework is open-sourced and available on GitHub[1] (Alexey et al., 2021). This release is based on the modified network with the CSPDarknet53 backbone and the YOLOv4 detector can be run by directly using the original Darknet-53 API through terminal commands. This GitHub repository was cloned to a computer running an NVIDIA GTX 1070 GPU such that the benefits of CUDA GPU acceleration would speed up all CV-related tasks (training and inference). It is important to already establish that the FISC topside PC does not have a dedicated GPU. To maximize processing speed during development, the topside PC was not used for any tasks that run the YOLOv4 model.

The model (kernel weights and configuration) included in the YOLOv4 repository is trained on the COCO dataset (presented in Section 2.6.4). The COCO dataset includes 91 different classes of objects (Lin et al., 2014), but none of these classes are aquatic animals. Therefore, a custom object detector must be trained to detect the desired objects in this project; Atlantic salmon. A guide in the YOLOv4 repository shows the step-by-step procedure on how to modify the backbone configuration files to prepare for training on custom datasets. This process requires two additional steps; 1) a labeled custom dataset and 2) a separate file with pre-trained weights. The pre-trained weights used for custom training are also trained on the COCO dataset and can be downloaded through the repository (*yolov4.conv.137*). Transfer learning, as presented in Section 2.6.4.1, is then used to train the network to detect the custom objects based on this pre-trained weights file. The training is initiated through a Darknet API terminal command.

There are a variety of open-source datasets for CV-related tasks, such as the Google Open Images Dataset[2] which includes over 9 million pre-labeled images on 600 different classes of objects. There are however no datasets (to the author's knowledge) that specifically include

[1] `https://github.com/AlexeyAB/darknet`

[2] `https://storage.googleapis.com/openimages/web/visualizer/index.html`

Atlantic salmon. Therefore, the datasets implemented in this project are purely based on custom labeling of images from Google, video frames from a colleague's Go Pro footage from a fish farm, and images captured with the FISC system itself. The *LabelImg*[3] tool was used to label the custom dataset, which supports the YOLO label format. This will be elaborated further in Section 4.4.

The completion of YOLOv4 training outputs a file containing the resulting kernel weights. This weights-file can then be exported and used in a YOLOv4 pipeline in any other framework, such as TensorFlow in Python.

3.2.2 TensorFlow and DeepSORT

TensorFlow is an open-source machine learning library with a native Python and C++ API. It was originally developed in 2011 by Google, but it was open-sourced in late 2015. It includes support for CUDA GPU acceleration and a large variety of machine learning models and architectures (Zaccone et al., 2017, pp. 30-32).

The DeepSORT MOT algorithm, presented in Section 2.6.5, has been integrated with a YOLO detection framework previously by *The AI Guy*[4]. Instead of attempting to ”re-invent the wheel” and construct the DeepSORT tracker from the bottom up, this open-source repository[5] was cloned from GitHub and modified to meet the specific requirements for this project.

The implementation in this repository allows for CUDA GPU acceleration, exporting resulting videos, and customizing parameters in the underlying YOLO detection framework (IoU and confidence threshold). The main modification which was needed for the use-case in this project was to extract the DeepSORT tracker information in every video frame timestamp which coincides with the timestamp of an acoustic Sector Focus ping. This allows for pixel coordinates and the ID of tracked targets to be used for the computation of 3D Cartesian coordinates, as presented in Section 2.4.

The original structure of the YOLOv4 weights file that the Darknet framework generates is not directly supported in the TensorFlow API. TensorFlow has its own standardized way of constructing models of artificial networks, which differs from other frameworks. Therefore, the YOLOv4 weights file must be converted to the correct format such that TensorFlow can interpret it and use them correctly. A model conversion script is included in the YOLOv4-DeepSORT repository, such that a custom-trained YOLOv4 model can be used in this open-sourced implementation.

[3] `https://github.com/tzutalin/labelImg`

[4] `https://github.com/theAIGuysCode`

[5] `https://github.com/theAIGuysCode/yolov4-deepsort`

3.2.3 Signal Processing

In the specialization project report, the data from the O_2/Temperature sensor and orientation sensor were not yet integrated into the data processing, but simply plotted and stored to a file together with the raw acoustic data. Additionally, the hydroacoustic signal processing was rudimentary at that time. This section will present the implemented processing of the above-mentioned sensors which has been performed during the master's thesis development, based on the theoretical principles presented in Chapter 2.

3.2.3.1 Orientation Sensor

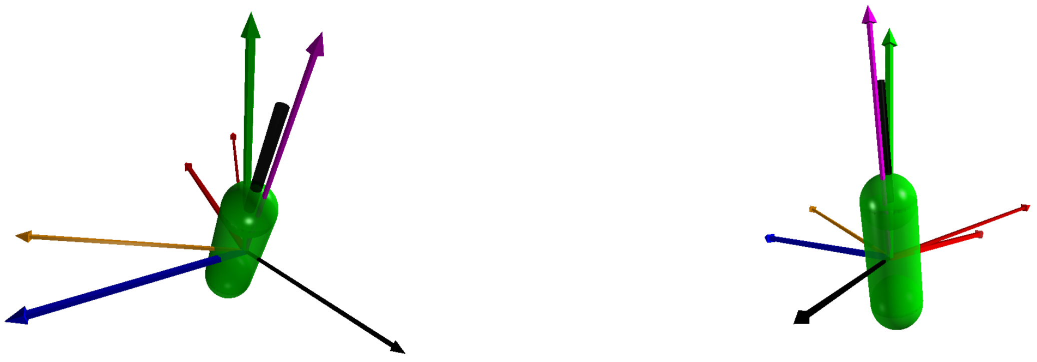
As detailed in Section 2.2, unit quaternions can be decomposed to Euler angles for an easier interpretation of orientation. The 9-DOF BNO055 orientation sensor is a System in Package (SiP) which has its own integrated MCU for running internal, but unspecified, sensor fusion software. It can therefore directly output quaternions at a rate of up to 100 Hz for absolute orientation estimation. The internal sensor fusion algorithms are claimed to give *"high robustness to magnetic distortions"* (*BNO055 Datasheet* 2020), which generally pose issues for sensors that rely on measuring the earth's magnetic field strength with a magnetometer. The BNO055 was chosen for the FISC prototype due to its positive reception in many embedded design forums (Arduino, Adafruit, etc.) and ease of implementation. The raw quaternions are stored to a file on the topside PC but also converted to Euler angles by using the operations shown in Equation 2.6 to show a live feed of the capsule's orientation during data acquisition.

In addition to roll, pitch, and heading, a directional vertical inclination estimation was also implemented. Vertical inclination describes how many degrees the sensor is offset from the fixed vertical axis, given from a combination of roll and pitch. The angle formed between the vertical axis in the fixed frame of reference and the vertical vector locked to the capsules orientation is calculated by applying the Pythagorean theorem to roll and pitch, i.e.:

$$\zeta = arctan(\sqrt{tan(roll)^2 + tan(pitch)^2}) \tag{3.1}$$

With the additional heading information from the sensor, this can be decomposed to show which compass direction the sensor is inclined towards. This parameter was implemented to determine the water current direction since the capsule suspended from the interface cable should naturally incline with the forces acting on it (water currents). The calculation of directional inclination is done through a vector decomposition. The vectors given from the Euler angles are used to find the horizontal projection of the capsule's vertical vector, which will always point towards the "source of inclination". Lastly, the angle between this horizontal projection and the north-aligned fixed frame of reference yields the source of inclination heading direction.

To better illustrate these calculations, Figure 3.4 shows two screenshots from a live 3D orientation visualization script made in Python during the orientation-specific software development. This testing script repeatedly fetches quaternions from the orientation sensor and visualizes the Euler-converted data in a 3D environment by using the VPython library. The thick red, green, and blue vectors are the fixed frame of reference, where red is in the north direction. The thinner red, orange, and purple vectors are fixed to the capsule orientation. The black vector is pointing towards the source of inclination. The capsule orientation is the same in both images, but the camera viewpoint is rotated to better illustrate that the black vector is the capsule's vertical vector horizontal projection. The angle formed between the black vector and the fixed frame of reference red vector gives the black vector's compass direction (anti-clockwise direction). This test script was used to verify the calculations before they were added to the acquisition and post-processing software.



(a) View 1.

(b) View 2.

Figure 3.4: FISC capsule 3D orientation visualization.

3.2.3.2 Acoustic Processing

The acoustic processing in the FISC system includes a discrete replica correlator, as presented in Section 2.1, and a target detection scheme implemented in Python.

The acoustic TX parameters (pulse length, center frequency, and bandwidth) are user configurable during acquisition and written to every generated file for post-processing, as previously presented. The replica correlator implementation is identical in both the acquisition software and post-processing software, where a custom Python function uses the TX pulse parameters to create a replica of the pulse. A Hamming window, implemented through Numpy's *numpy.hamming* function, tapers the start and end of the pulse's amplitude to minimize sidelobes during transmission and correlation, as presented in Section 2.1.1.

The replica correlation process itself is implemented through the Python library *Scipy*, which includes many built-in functions for generic signal processing. The *scipy.signal.correlate* function takes two array inputs (raw acoustic RX data and TX pulse replica in this case) and outputs the cross-correlation. The correlation is performed with the FFT-method presented in Section 2.1 to maximize computation speed. The envelope of the resulting signal is extracted as the final step in the correlation procedure.

The envelope signal is then passed through a CA-CFAR detector (presented in Section 2.1.2), to find possible target echoes. The implemented CA-CFAR Python function is a modified version of one accessible online[6]. Initially, the output from the CA-CFAR detector was supposed to be the last step in the acoustic target detection processing. For reasons which will be elaborated on later, an additional peak detector was added as a supplementary stage to ease the optical and acoustic sensor fusion-related calculations. The *Scipy* library has a dedicated function for finding peaks in signals ($scipy.signal.find\_peaks$), which allows for numerous input parameters to tailor the detector to the specific data. It is however important to note that this final peak detection stage was used as a crude ”max peak” detector on the output of the CA-CFAR detector.

Figure 3.5 shows a raw acoustic sample as well as the final output after the aforementioned processing steps. The peaks from echoes within the acoustic deadzone are ignored in the detector, and only the largest CA-CFAR detection output is fed through to the fusion processing.

The FISC software visualizes data in two different ways, where a standard plot of the processed data is mostly used for acoustic analysis (live and in post-processing), while a polar plot is used to visualize more sensor readings during acquisition or replays of recorded data. The standard plot is essentially the bottom plot in Figure 3.5. An example of the polar plot is shown in Figure 3.6. Here, the matched filter output amplitudes for all 8 echosounder sectors are color mapped to visualize the presence of targets relative to the capsule orientation. The four radial lines in the outer perimeter of the plot are compass heading indicators, where the red line is the north direction. In the polar plot example, Sector 3 is pointing north. The orientation sensor is mounted on the internal custom PCB such that Sector 4 is aligned with the sensor's native north direction. Additional sensor information is visualized below the polar plot, where there are four measurements for orientation readings since the orientation sensor is sampled between every acoustic ping.

[6] `https://tsaith.github.io/detect-peaks-with-cfar-algorithm.html`

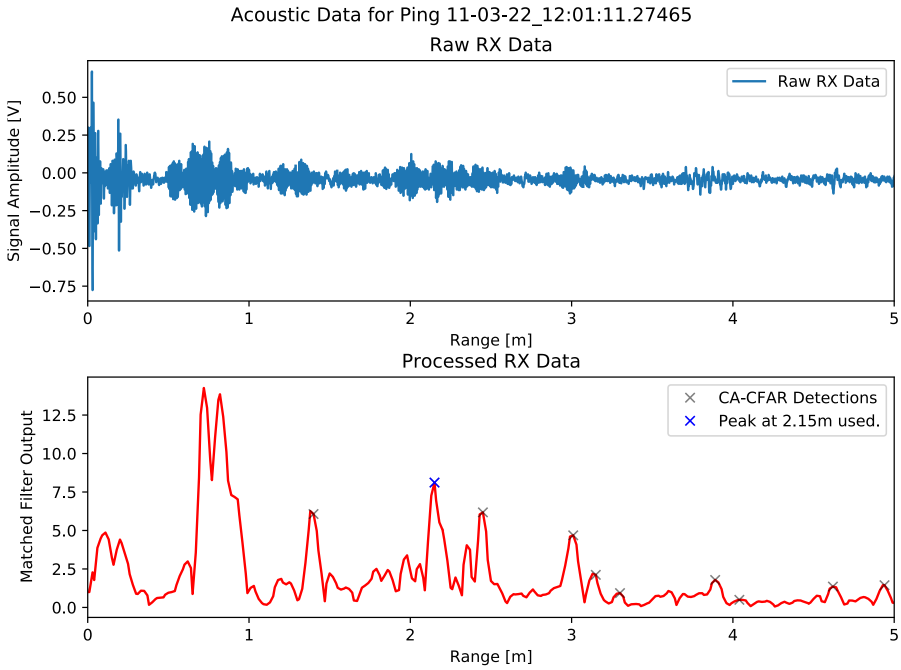


Figure 3.5: FISC acoustic processing pipeline example.

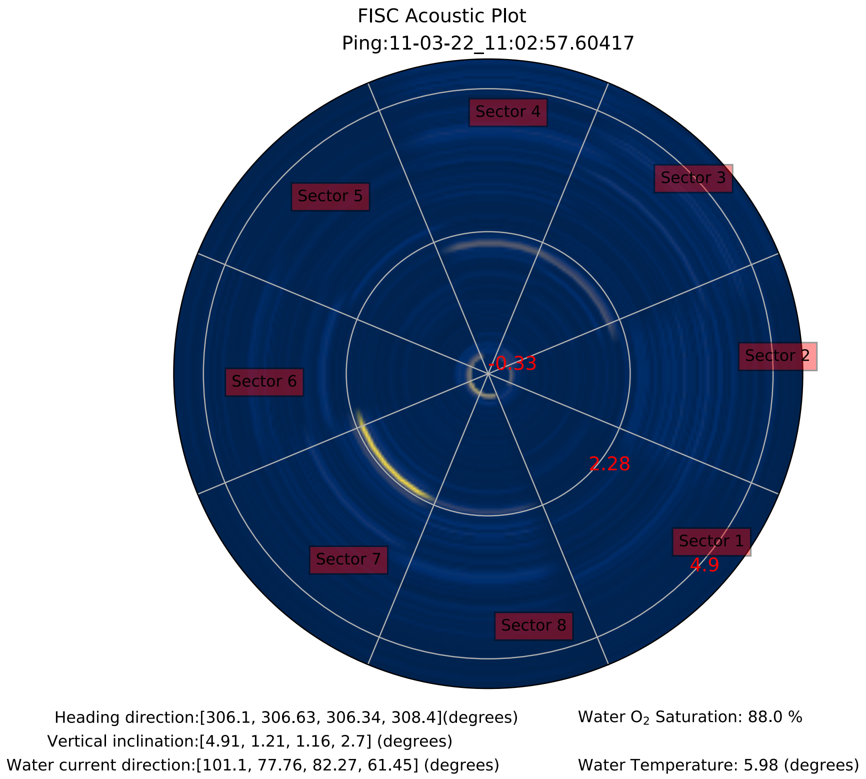


Figure 3.6: FISC polar plot example.

3.2.4 Aquatic Environment Quality Score

Based on the O_2 and water temperature thresholds presented in Section 2.5.2, two mathematical functions were defined to create a combined water quality score grid. The temperature score function is a bell curve with a maximum amplitude of one with its peak occurring at 12.5 °C, which according to Table 2.1 is the optimum temperature for maximal growth and food intake for Atlantic salmon. The O_2 saturation score function is a combination of an exponential function and a fixed value, where the fixed value covers the range above optimum in the cited work. Figure 3.7 shows the two resulting functions for both aquatic parameters.

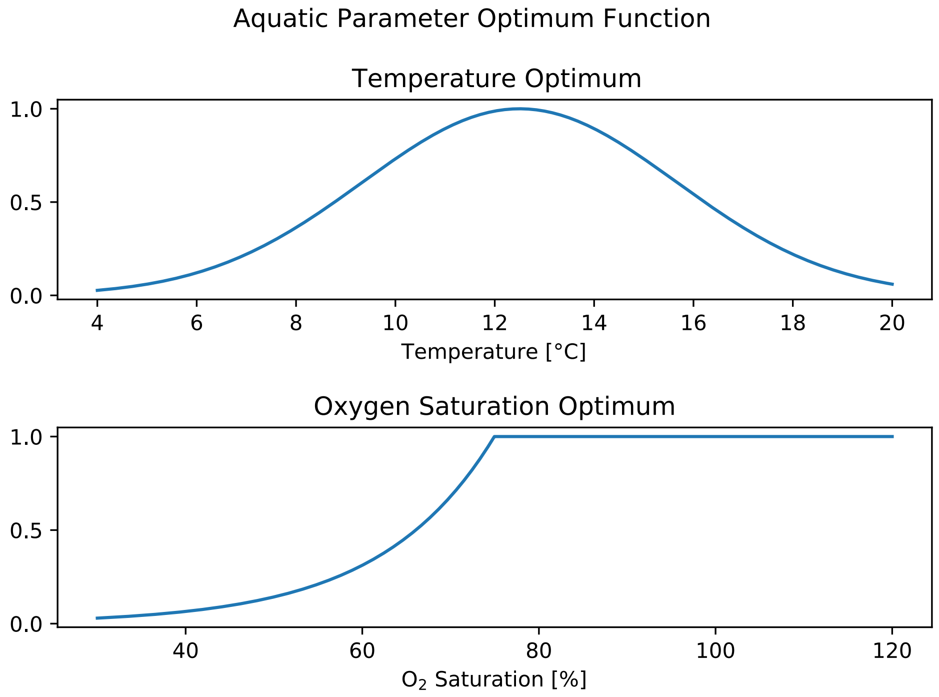


Figure 3.7: Custom aquatic environment score functions.

The multiplicative combination of these two score functions can be used to generate a grid that yields a combined score ($\in [0,1]$) based on one temperature and O_2 sample, as shown in Figure 3.8. This grid was created with the *numpy.meshgrid* function, where a custom colormap ranging from red to green was defined. The resulting water quality score is mainly implemented in the depth profile visualizations in the post-processing, but can also be presented live during acquisition. These two aquatic parameters often vary with depth and over time, and it is generally desirable to present them as a function of depth for the fish farmers to have more knowledge about the aquatic environment before/during feeding.

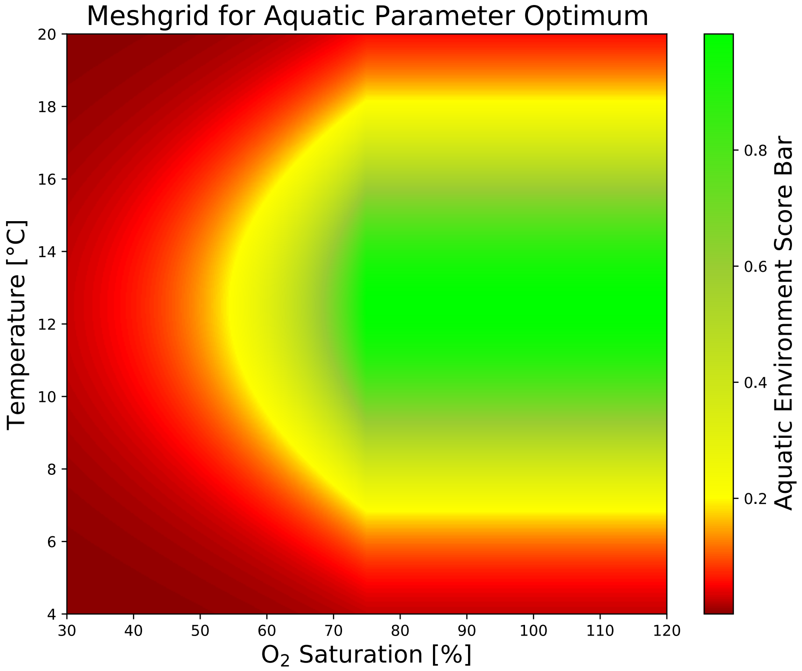


Figure 3.8: Custom aquatic environment score heat map.

3.2.5 FISC Post-processing Pipeline

This section will stitch together the aforementioned implementations and present the resulting processing pipeline which is used to both generate and process sensor fusion data acquired with the FISC prototype. The complete FISC post-processing Python repository with a selection of usable data can be downloaded (Ericsson, 2022), enabling the readers to run the software themselves. A complete installation guide with usage instructions is found in the repository.

There are two main scripts in the post-processing pipeline. One script is for visualizing and saving plots of data. The other script is dedicated to optical and acoustic sensor fusion. They both take additional call arguments to specify which data to process, what time window to extract data from, which field test data to use (detailed below) and how the data should be visualized. The two scripts and their functionalities are shown in Table 3.2.

viewSavedData.py	**fusion.py**
- Plot IMU data over time (orientation values or water current direction estimation). - Plot O_2 and temperature-data (over time or depth-profile). - Plot acoustic data (Sector Scan or Sector Focus) over time to replay acquisition session. - Plot optical and acoustic data synchronized in time to evaluate if some sequences are viable for sensor fusion. - Export any of the above plots.	- Run the custom YOLOv4-DeepSORT model on raw videos to generate and export CSV-files containing tracker information. Optically tracked fish and their respective bounding box parameters are extracted at all video-frames coinciding with acoustic pings. - **Process Data** · Use the acquired FISC data and generated YOLOv4-DeepSORT tracker information files to perform time-synchronized plotting (optical and acoustic) with individual tracking. · Estimate size and swimming speed of individuals based on optical and acoustic sensor fusion (calculations from Section 2.3), one individual in focus per sequence. Assumes that the single acoustic detection output from acoustic processing pipeline is an echo from the current DeepSORT ID track in focus.

Table 3.2: FISC post procesing main scripts with details.

The specific Python function calls used to generate all plots shown in Chapter 4 will include a footnote with their specific arguments.

3.3 Details on the Initial Field Test in Rørvik

Since the fall of 2020, Norbit Subsea AS has been involved in a project called CrowdGuard[7], led by Birger Venås at SINTEF Ocean AS. The CrowdGuard project is a consortium of several technology-/ and aquaculture-related companies, where the project's purpose and goal were to discuss and develop technological solutions to improve the crowding process in salmon aquaculture.

[7] `https://www.sintef.no/en/projects/2018/crowdguard/`

The crowding process in itself is not a main focus of the FISC system, but it still received attention from those involved in the consortium. This attention resulted in the possibility to test the prototype in its true environment, and Sinkaberg Hansen AS allowed dispensation of their fish-farming sites.

The initial field test took place on the 16th of February in Marøya, Rørvik, outside Sinkaberg Hansens headquarters and factory. The factory is located by the water, where they have five ”waiting-pens” which the salmon are in during the very last stage of their life cycle before slaughter (<6 days (Lovdata, 2008, §54)). Figure 3.9 shows the Sinkaberg Hansen factory, with a pin on the pen where the FISC prototype was tested.

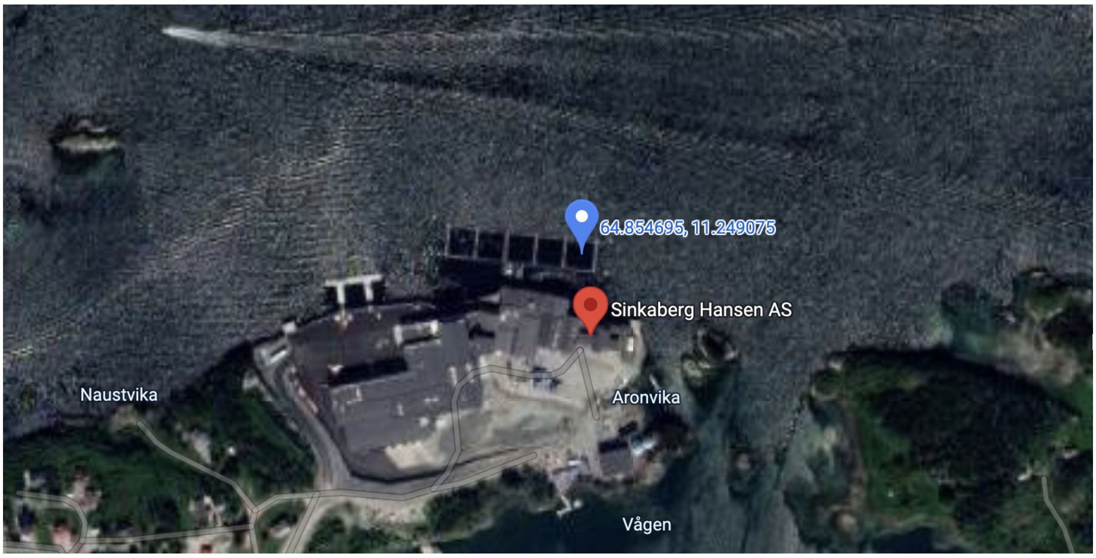


Figure 3.9: Sinkaberg Hansen AS factory.

The author and three employees from Norbit Subsea AS joined the trip, to assist and perform tests with other sonar equipment. A total of 4 hours were spent acquiring data with the FISC system, where different system configurations and capsule locations were tested. This will be detailed further in Section 4.2.

3.4 Details on the Second Field Test near Frøya

Due to a few challenges with the acquired data from the initial field test (detailed later in Section 4.2), a second field test was also completed. This test took place on March 11, at the Rataren fish farming site near Frøya. Rataren [8] is a full-scale fish farm that is partially operated by SINTEF ACE and SalMar ASA. SINTEF ACE has several locations where commercial fish farming takes place under research concessions and SalMar ASA controls the commercial aspect of

[8] https://www.sintef.no/alle-laboratorier/ace/

the operations at the Rataren site. These concessions allow SINTEF to facilitate research and testing of products under development in a full-scale environment, which was made possible in this project through Birger Venås.

The Rataren site is shown in Figure 3.10, where Rataren II consists of the seven fish pens on the right side. The specific testing pen is pinned on the image. Birger Venås organized and joined the trip, where a boat with a skipper (Terje Bremvåg, SINTEF Ocean AS) was acquired for transportation and assistance. Arild Søraunet (project technical supervisor, Norbit) and Guttorm Lange (senior aquaculture advisor, Norbit) joined the field trip to assist and assess the FISC system functionality.

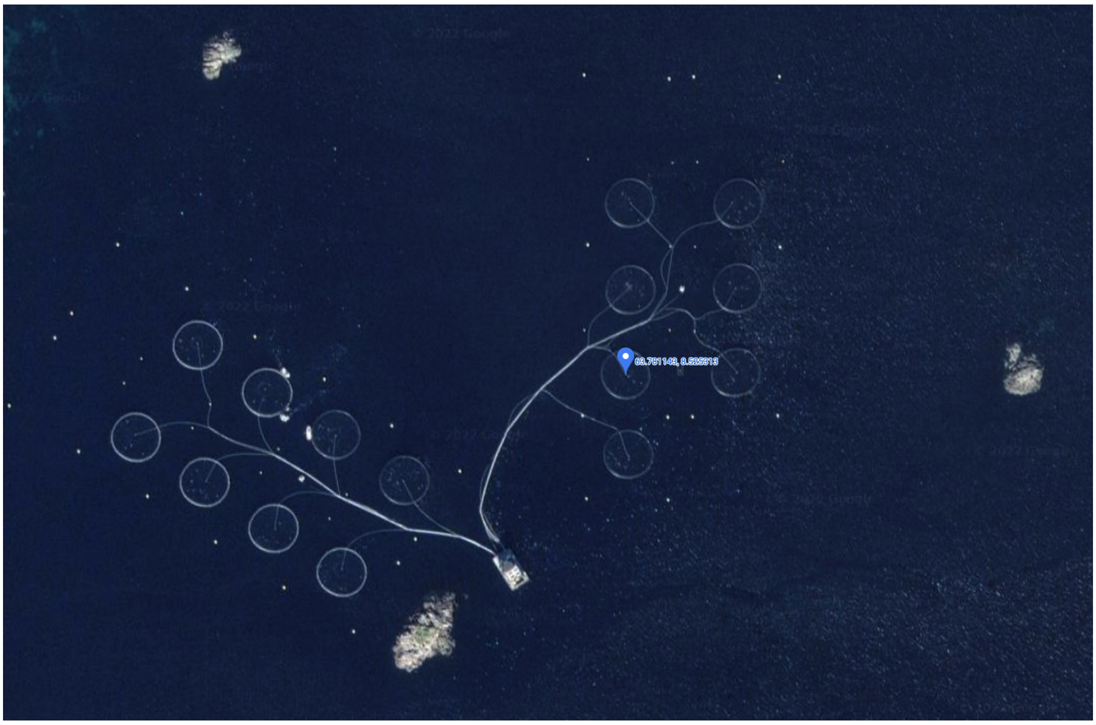


Figure 3.10: Rataren site near Frøya (SINTEF ACE / SalMar ASA).

Chapter 4

Results

4.1 Acoustic Vertical Beam Pattern Characterization

Ericsson (2021) presented a characterization of the horizontal beam patterns for the FISC prototype acoustics. It is equally important to characterize the vertical beam pattern to better understand the acoustic coverage in the vertical domain, especially since the TX and RX are separated by 0.3 m.

In itself, the TX and RX separation distance poses a possible issue. The further separated they are, the larger the radial acoustic deadzone between them will become. The acoustic deadzone consists of the volume of water between the TX and RX that the main lobe acoustics never propagate through, due to the limited beamwidth of both transducers. This is illustrated in Figure 4.1. The following calculations assume the standard method of using the 3 dB beamwidth as the acoustic coverage descriptor. Hence, the ”outer parts” of the vertical main lobe, as well as vertical sidelobes of the TX, will to some extent fill this acoustic void. Since only the main lobe acoustics give ”valuable information” in the implemented acoustic processing, echoes received within the deadzone range are ignored in the detector.

Ericsson (2021) also detailed the fundamental theory behind piezoelectric ceramics, hydroacoustic transducers, and their propagation characteristics. To summarize what is essential and re-used in the following, it is the geometrical shape and active area of the transducer which determine its beam pattern. The relationship between beam width and transducer length in the same axis is inverse. The longer the active area is, the narrower the resulting main lobe becomes. The specific geometrical characteristic in focus is the height of the receiver elements since this governs the acoustic deadzone. Covering a portion of the receiver elements with a material that has a low acoustic impedance, e.g. a layer of cork composite, limits the active area of the receiver elements. Hence, this will allow for modifications of the beamwidth. Based on expe-

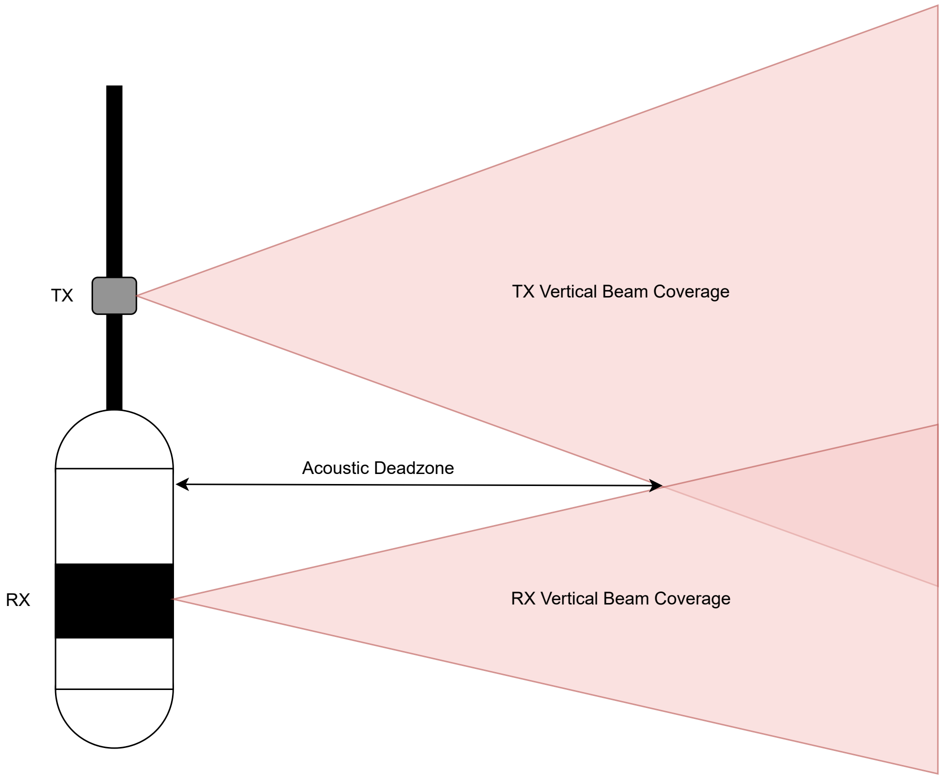


Figure 4.1: Illustration of acoustic deadzone in the FISC prototype.

rience from engineers at Norbit Subsea AS, this procedure works well for beam characteristic modifications, called *shading*.

By using the same beam pattern simulation script applied by Ericsson (2021), the vertical dimensions of the ceramics are input to visualize the vertical beams. Figure 4.2 shows the TX vertical beam pattern with a 3 dB beamwidth of 12.07 °. Figure 4.3 shows the RX vertical beam patterns for a set of different lengths in the vertical axis. The installed FISC RX Sector ceramics are composed of two 10 mm elements in series vertically, meaning 20 mm height is the original configuration. The 3 dB beamwidths for 20 mm, 15 mm and 10 mm element heights are 7.75 °, 10.63 ° and 15.68 °, respectively.

Estimating the radial acoustic deadzone is then done through a trigonometric analysis, where we want to find x in Figure 4.4.

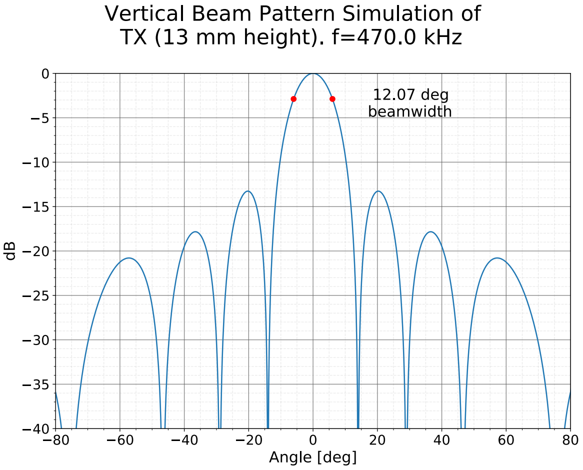
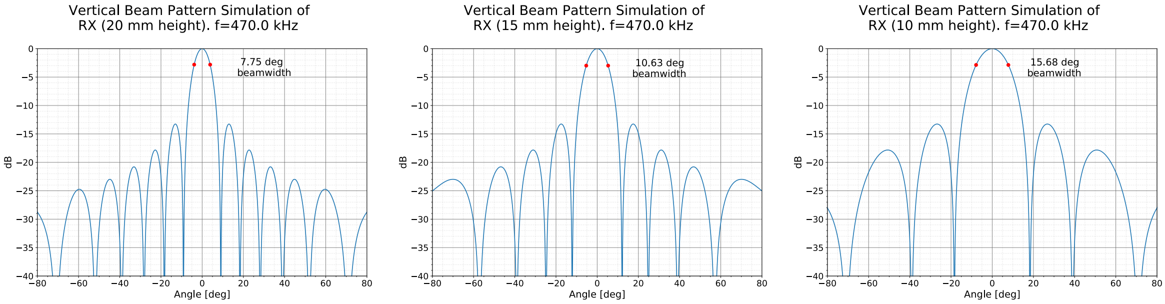


Figure 4.2: FISC TX vertical beam pattern.



(a) 20 mm (Original).

(b) 15 mm (5mm shading).

(c) 10 mm (10 mm shading).

Figure 4.3: FISC RX vertical beam patterns for different heights and shadings.

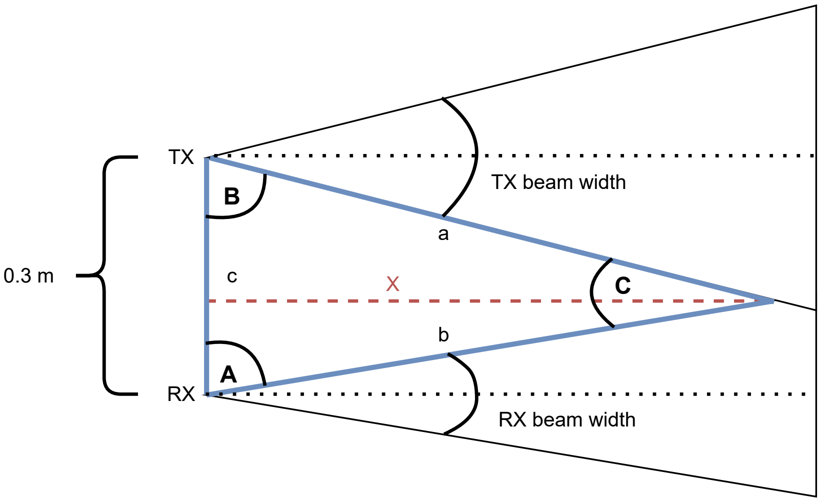


Figure 4.4: FISC acoustic deadzone geometry breakdown.

The distance between the TX and RX center is known (0.3 m). The angles A and B are given from the respective vertical beamwidths, and the angle C is then easily found, i.e.:

$$
\begin{aligned}
A &= 90^\circ - \frac{RX_{beam\ width}}{2} \\
B &= 90^\circ - \frac{TX_{beam\ width}}{2} \\
C &= 180^\circ - A - B
\end{aligned}
$$

The law of sines, which apply to any triangle, state that:

$$\frac{a}{sin(A)} = \frac{b}{sin(B)} = \frac{c}{sin(C)}$$

Now, we arbitrarily define b with the law of sines from known angles and sides:

$$
\begin{aligned}
\frac{b}{sin(B)} &= \frac{c}{sin(C)} \\
&\Downarrow \\
b &= \frac{c \cdot sin(B)}{sin(C)}
\end{aligned}
$$

Lastly, the unknown length x can be defined from simple trigonometry with the side b, i.e.:

$$
\begin{aligned}
x &= b \cdot sin(A) \\
&\Downarrow \\
x &= \frac{c \cdot sin(A) \cdot sin(B)}{sin(C)}
\end{aligned}
\tag{4.1}
$$

By using Equation 4.1 and the three different RX beamwidths from Figure 4.3, a set of acoustic deadzones are calculated:

RX Vertical Height	Radial Acoustic Deadzone (x)
20 mm (No shading)	1.73 m
15 mm (5 mm shading)	1.51 m
10 mm (10 mm shading)	**1.23 m**

Table 4.1: Radial acoustic deadzone with different RX shadings.

From these calculations, a cork layer was wrapped around the FISC capsule, where a shading of 10 mm was chosen. It is important to note that this alteration was performed in preparation for the second field test, so the initial field test had the original 1.73 m deadzone.

The vertical acoustic coverage with this modification was then characterized to determine the total resulting acoustic beam shape in the vertical domain. A test tank facility operated by Norbit Subsea AS was used for this characterization, where Figure 4.5 illustrates the setup.

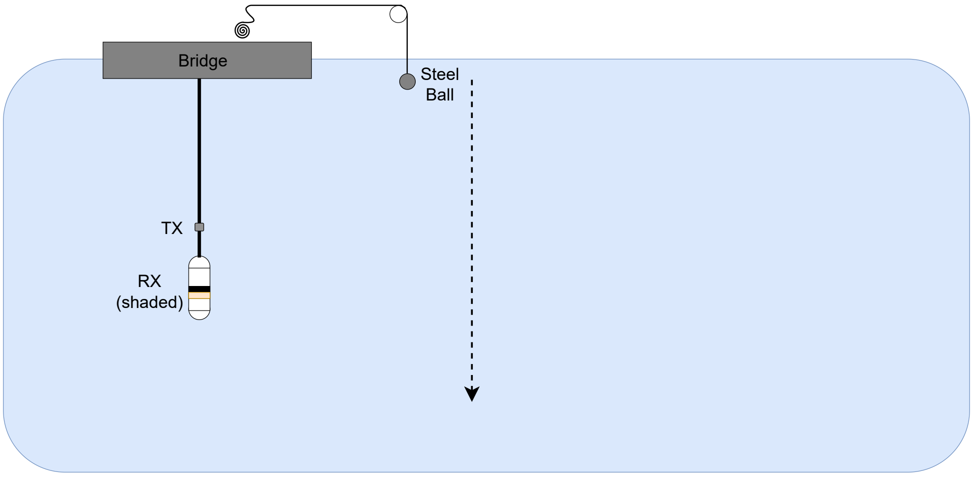


Figure 4.5: Vertical acoustic beam characterization setup.

The FISC system was placed in the middle of the water column in the tank and set to continuously operate in the Sector Focus mode, where Sector 4 is directed towards the steel balls path. The echoes received from the steel ball are used to measure the received acoustic strengths. The ball is lowered slowly with a constant velocity by using a drill. The timestamps of when the ball is at 0 m and 2.3 m depth are noted. This enables the determination of the angle between the acoustic center and the current depth of the ball at every ping, and the received echo strengths are used to visualize the relative beam at every angle. The python script *verticalAperture.py*, found in the thesis repository, was used for these calculations. Figure 4.6 shows the processed data from this test, where it is observed that the estimated total beam width is approximately 22.5 °. This will be discussed further in Chapter 5 due to a few peculiarities with the obtained result. Nevertheless, this estimated beam width was used to compare the vertical coverage of the acoustics and the camera, seen below in Section 4.5.

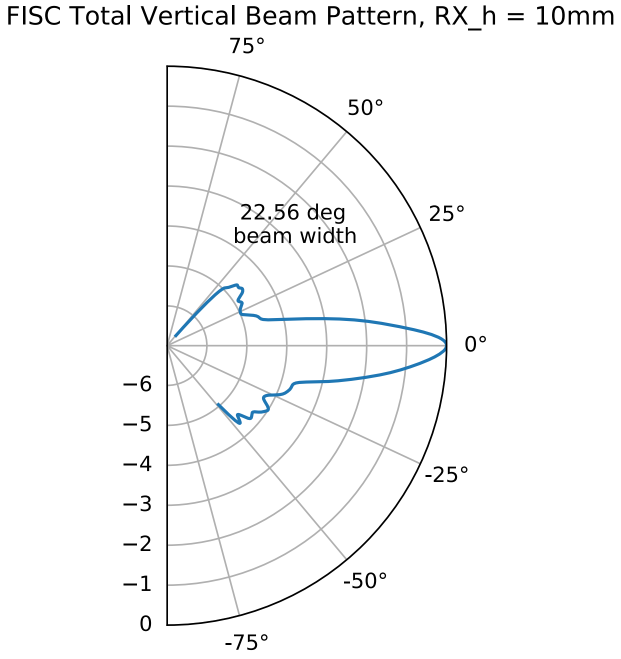


Figure 4.6: Characterization result of FISC vertical beam pattern.

4.2 Initial Field Test Data Acquisition

The initial field test at Sinkaberg Hansen AS in Rørvik was mainly a verification of system functionality in an uncontrolled environment, but also the initial data acquisition for post-processing development. Since the system had not been tested outside of office testing facilities before this, a detailed list of specific tests and checks to perform was created prior to the trip.

A sound velocity profiler from Norbit Subsea was brought along to Rørvik to acquire sound velocity data at every depth in the water column. This data is used in the post-processing of the acoustics to get precise distance measurements to objects. The following list summarizes the most important parts of the initial field test acquisition plan:

1. Sanity check:
   - Verify correct timing and compass heading.
   - Acquire data in air and verify video quality.
2. Acquire data under water and verify that the acoustics function and camera lighting conditions are adequate.

3. If no issues have occurred, acquire data at ≈2 m depth for 10-15 minutes with the following parameters:

    - TX Pulselength: 450 μs
    - TX Center Frequency: 470 kHz
    - TX BW: 80 kHz
    - Max range: 15 m

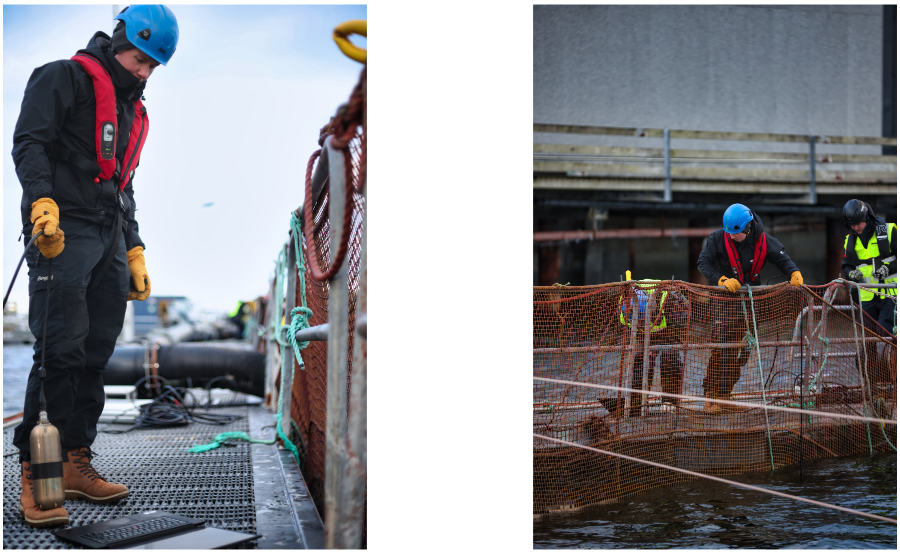
    Plot some of this data to assess quality. Final verification step before acquisition.

4. Profile Acquisition

    - Use external sound velocity profiler to fetch sound speed as a function of depth (not within pen).
    - Mount FISC capsule in a way that allows for lowering in depth freely.
    - Acquire data with previous parameters for 1 minute at every meter mark on cable to generate a depth profile.

5. Repeat step 3 with various parameters (max range and TX BW) at different depths.

After verification of system functionality in air, the FISC capsule was lowered into the water column manually to verify the acoustics and video image quality. Figure 4.7 shows two images taken during system verification in air and water.

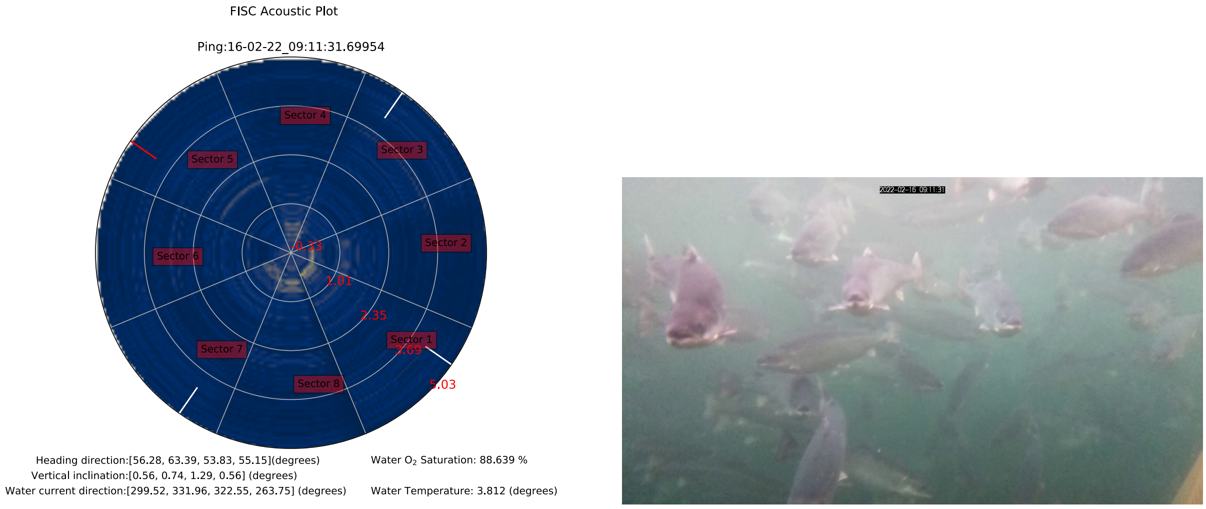


(a) Verification in air.

(b) Verification in water.

Figure 4.7: Images from initial field test system verification process.

Figure 4.8 shows a Sector Scan plot and a video frame from the FISC capsule camera which verified system functionality in water before the main data acquisition was initiated.



(a) FISC Sector Scan validation plot.

(b) FISC camera validation frame.

Figure 4.8: Validation data before initial field test acquisition started.

A rope was then tied in a large loop on two sides of one corner in the waiting pen to function as a pulley, enabling the capsule to be pulled further towards the radial center of the pen. The interface cable was fed through a small loop in the rope to easily allow for changes in depth by lowering or pulling the cable. An illustration of the installation can be seen in Appendix H, Figure H.1a.

The FISC data acquisition software was then set to continuously gather data from all sensors (acoustic, optical, orientation, and aquatic environment). The system swapped between Sector Scan and Sector Focus mode every 15 seconds.

Lowering the capsule and noting the time at every depth enabled a depth profile plot to be generated after acquisition. The depth profile measurements were taken between 09:48 and 09:52, and the resulting O_2 and temperature profile with their individual color-graded quality score is depicted in Figure 4.9[1]. The combined water quality score is based on extracting the heat map coordinate value from the two separate measurements, as presented in Figure 3.8, Section 3.2.4. The sound velocity profile from the initial field test can be seen in Figure J.1a, Appendix J.

[1] Exact function call: python3 viewSavedData.py --o2temp --profile

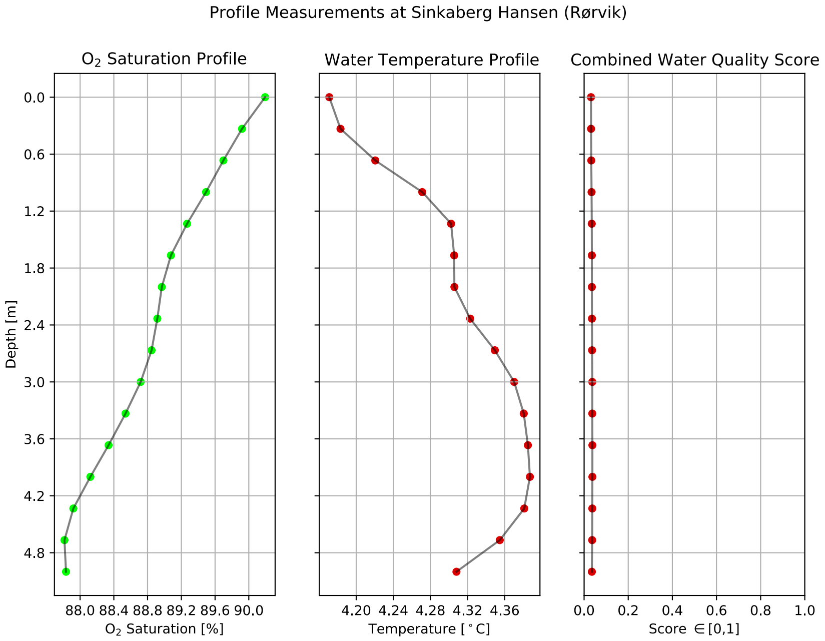
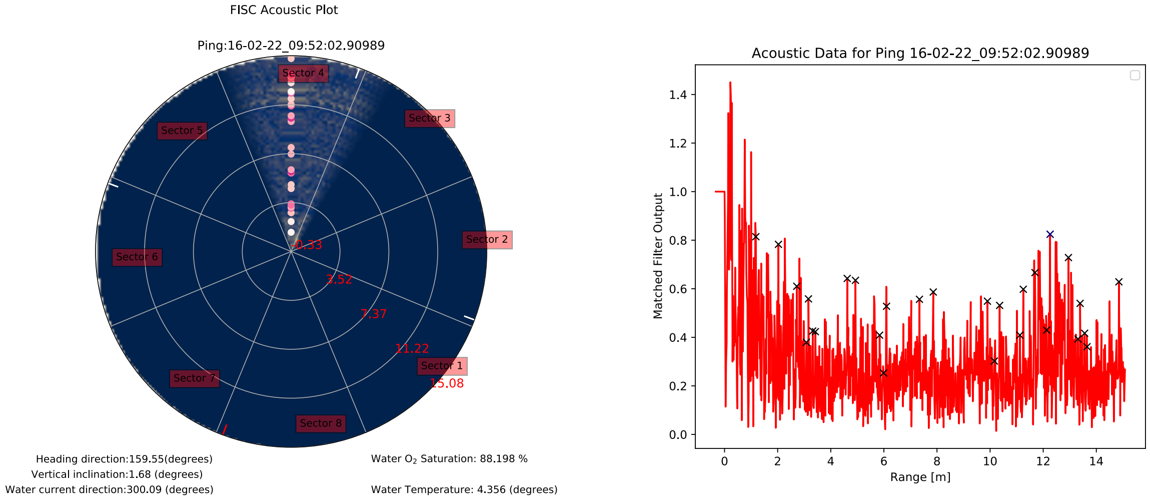


Figure 4.9: Water quality depth profile at Rørvik.

After the profile acquisition was completed, the system was set to gather data in the same installation from 10:00 until 11:36, as the capsule depth was altered in 15-20 minute intervals. Figure 4.10 shows an acoustic ping in the Sector Focus mode with the CA-CFAR peak detector output enabled, which shows the presence of many targets (peaks in signal).



(a) Sector Focus polar plot.

(b) Matched filter output plot.

Figure 4.10: Acoustic ping with CA-CFAR detector output enabled.

Since the fish only spend a maximum of 6 days in the waiting pen before slaughter, the legal stocking density requirements are less strict in this production phase (Lovdata, 2008, §25).

Therefore, most of the acoustic data from the initial field test was saturated beyond separability due to the large number of individuals per volume of water. Furthermore, both the O_2/Temperature sensor and orientation sensor were sampled between every acoustic ping in the Sector Focus mode at this time. It wasn't until after the post-processing development began that it became apparent that the acoustic ping rate is heavily limited by these sensors.

It takes a maximum of **0.4 s** to sample the orientation sensor and anywhere from **0.5 s** to **1.5 s** to sample the O_2/Temperature sensor based on tests performed in the lab after the initial field test. This delay can decrease the maximum acoustic ping rate by nearly 2 Hz, which reduces the possibility of acoustically tracking targets. This issue was addressed in the second field test.

Figure 4.11 shows a water current direction estimate over time[2], based on the directional vertical inclination calculation presented in Section 3.2.3.1. The time duration covers the entirety of the field test when the system was mounted in the aforementioned rope installation, with approximately 20 orientation samples per minute (0.33 Hz).

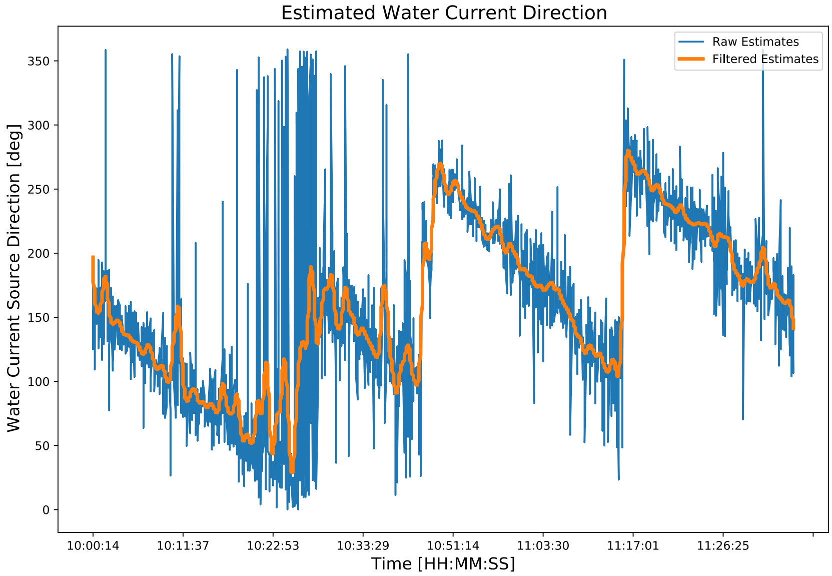


Figure 4.11: Water current source direction during initial field test.

[2] Exact function call: python3 viewSavedData.py --imu

4.3 Second Field Test Data Acquisition

Due to both high fish density and limited acoustic ping rate during the initial field test, a second field trip was completed to acquire data that would be more feasible for optical and acoustic sensor fusion development.

The Sector Focus routine was updated to sample only acoustic and visual data, which resulted in a maximum attainable acoustic ping rate of approximately 6 Hz (limited by the processing time of data). To minimize synchronization errors in post-processing, the generated timestamps on the saved videos were given decimal point precision. Lastly, the RX vertical aperture was halved to decrease the acoustic deadzone to 1.23 m, as mentioned in Section 4.1, Table 4.1. This modification is observed previously, in Figure 3.3, where cable ties were used to fasten a 2 mm layer of cork composite around the receivers. Gaffer tape was wrapped around to avoid any sharp edges to avoid possible injuries to colliding fish.

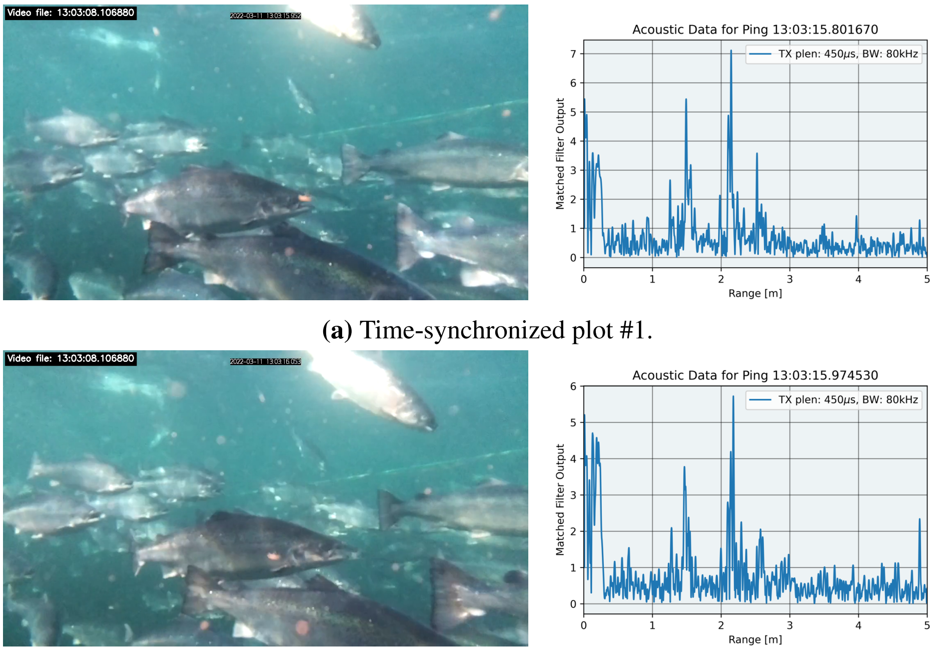
The main focus of this field trip was to gather as much data as possible in the Sector Focus mode, to obtain a lot of data for sensor fusion development. An installation similar to the one in the initial field test was made possible through assistance from Terje (the boat skipper). A buoy tied between two ropes served as a floater which the interface cable was fed through. The other ends of the ropes were fastened at different pen perimeter locations to move the FISC capsule towards the center of the circular pen. An installation illustration can be seen in Figure H.1b, Appendix H. An underwater image of the FISC capsule in the pen water column captured during the second field test can be seen in Appendix I.

Figure 4.12 shows two subsequent time-synchronized optical and acoustic samples from the Sector Focus mode during the second field test, generated with the *viewSavedData.py*[3] FISC post-processing script. The time delta between the two samples is ≈0.173 s, which equates to a ping rate of nearly 6 Hz. The most important observation to draw from this example is the dynamic consistency in the two plots, where several of the echoes (peaks) are present in both acoustic pings. This is an essential requirement for tracking individuals acoustically which was not distinguishable at all in the acoustic data from the initial field test.

Identical to the procedure in the initial field test, an O_2/Temperature depth profile was performed at the Rataren II site, from 12:50 to 13:00. This is shown in Figure 4.13, with the same water quality score determination procedure[4]. The sound velocity profile for this field test can be seen in Figure J.1b, Appendix J.

[3] Exact function call: python3 viewSavedData.py --syncPlot --sf --show --ace --savePlots --start 13:03

[4] Exact function call: python3 viewSavedData.py --o2temp --profile --ace



(a) Time-synchronized plot #1.

(b) Time-synchronized plot #2.

Figure 4.12: Subsequent time-synchronized optical and acoustic samples.

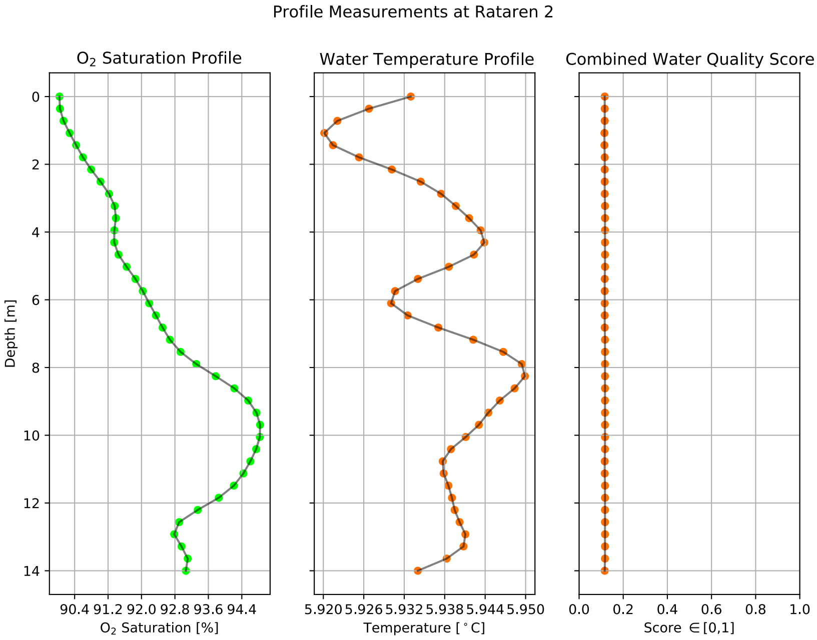


Figure 4.13: Water quality depth profile at Rataren II.

Figure 4.14 shows the water current direction estimates for the second field test[5]. Since the orientation sensor wasn't sampled during the Sector Focus mode during this test, and the system ran in this mode exclusively between 11:04 and 12:31 (and after 13:00), no orientation data is

[5] Exact function call: python3 viewSavedData.py --imu --ace

available at these times. The window without orientation data is labeled with the red vertical line.

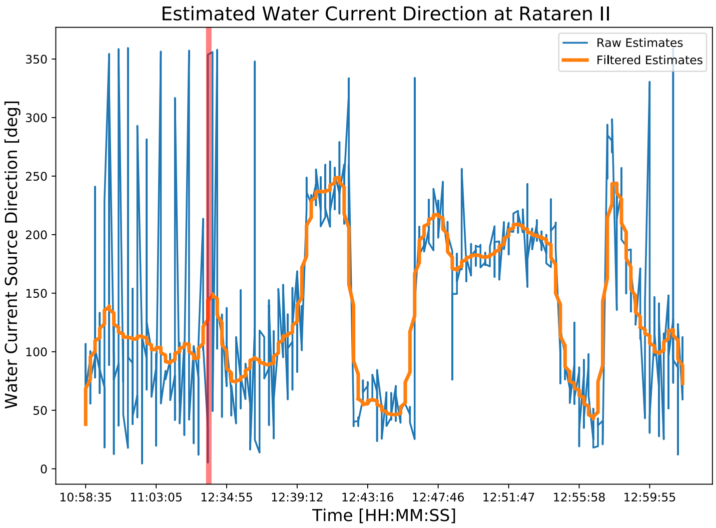
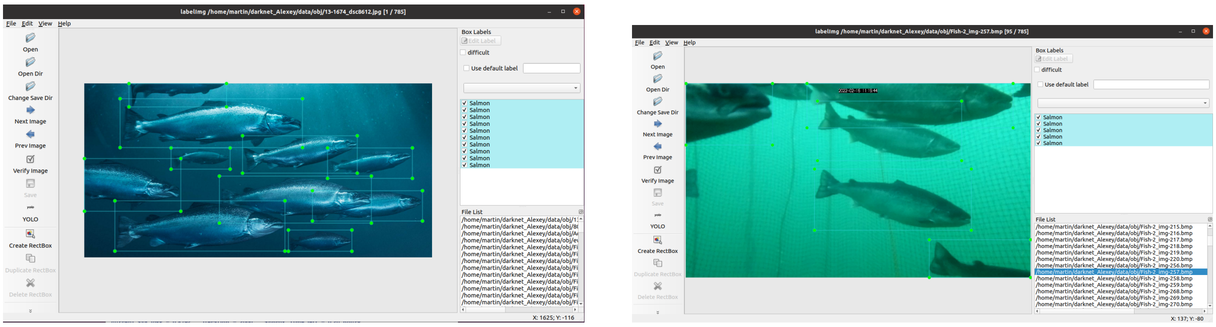


Figure 4.14: Water current source direction during second field test.

4.4 Training of Custom YOLOv4 Salmon Detector

As detailed in Section 3.2.1, a YOLOv4 detector with manually labeled data was used in the Deep learning pipeline in the FISC post-processing software. The tool *LabelImg*[6] was used to manually label images of Atlantic salmon in a variety of lighting conditions and orientations. Figure 4.15 shows a screenshot during custom labeling, where an image from Google and the initial field test are being labeled.



labelImg /home/martin/darknet_Alexey/data/obj/13-1674_dsc8612.jpg [1 / 785]

□ difficult

□ Use default label

☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon

File List

/home/martin/darknet_Alexey/data/obj/1:
/home/martin/darknet_Alexey/data/obj/8(
/home/martin/darknet_Alexey/data/obj/A(
/home/martin/darknet_Alexey/data/obj/ev
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi
/home/martin/darknet_Alexey/data/obj/Fi

X: 1625; Y: -116

(a) Custom labelling of Google image.

labelImg /home/martin/darknet_Alexey/data/obj/Fish-2_img-257.bmp [95 / 785]

2022-02-16 11:19:44

□ difficult

□ Use default label

☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon
☑ Salmon

File List

/home/martin/darknet_Alexey/data/obj/Fish-2_img-215.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-216.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-217.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-218.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-219.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-220.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-256.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-257.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-258.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-259.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-268.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-269.bmp
/home/martin/darknet_Alexey/data/obj/Fish-2_img-270.bmp

X: 137; Y: -80

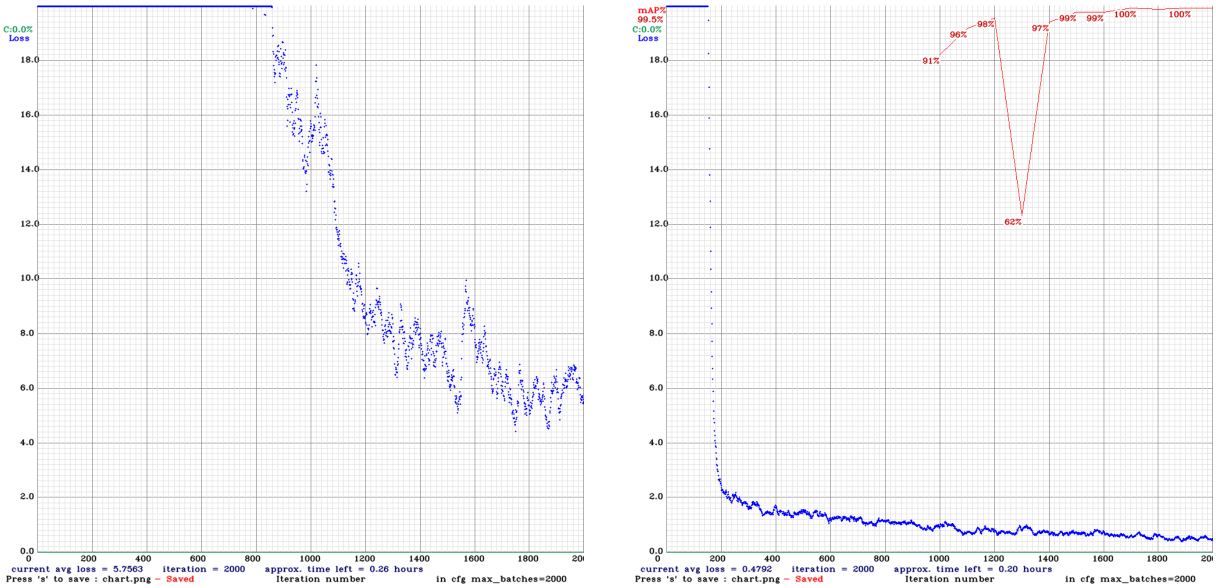
(b) Custom labelling of FISC video frame.

Figure 4.15: LabelImg screenshots during custom labeling of YOLOv4 training data.

The custom YOLO detector was trained three separate times, where only the last trained model was used in the post-processing pipeline. The first completed training gave unsatisfactory re-

[6] `https://github.com/tzutalin/labelImg`

sults and was discarded. The first training was performed early in the development to mainly gain more understanding of the Darknet API and YOLO detector framework in general. After more images of Atlantic salmon were harvested from GoPro footage supplied by a colleague, as well as video frames from the initial field test, a total of 184 manually labeled images were used for training the detector. A third training was performed based on the following statement in the official YOLOv4 repository[7]: (it is) *”desirable that your training dataset include images with non-labeled objects that you do not want to detect - negative samples without bounded box (empty .txt files) - use as many images of negative samples as there are images with objects”*. Therefore, 184 images without Atlantic salmon were supplemented to the dataset, resulting in a total of 368 images. The results from the first and last model training are depicted in Figure 4.16.



(a) First YOLOv4 model training result.

(b) Third (final) YOLOv4 model training result.

Figure 4.16: YOLOv4 custom model training results.

It is important to identify that the mAP parameter was not enabled in the first training routine. This extra parameter during training simply computes the mAP, as detailed in Section 2.6.3, at every 100 iterations from the halfway point to show the model’s accuracy. The resulting weight files are not different in that sense. The loss curves are very different in the two training routines, where both the resulting loss value and convergence curve are far better in the last model. The mAP during the final training is seen to achieve 99.5 % towards the end of the training routine. All three training results can be seen in Appendix G.

A video frame export generated with the final YOLOv4 model and Darknet API is shown in Figure 4.17. This verified that the model indeed detects Atlantic salmon with reasonable confidence scores, shown on the bounding box labels (≈98 %).

[7] `https://github.com/AlexeyAB/darknet`

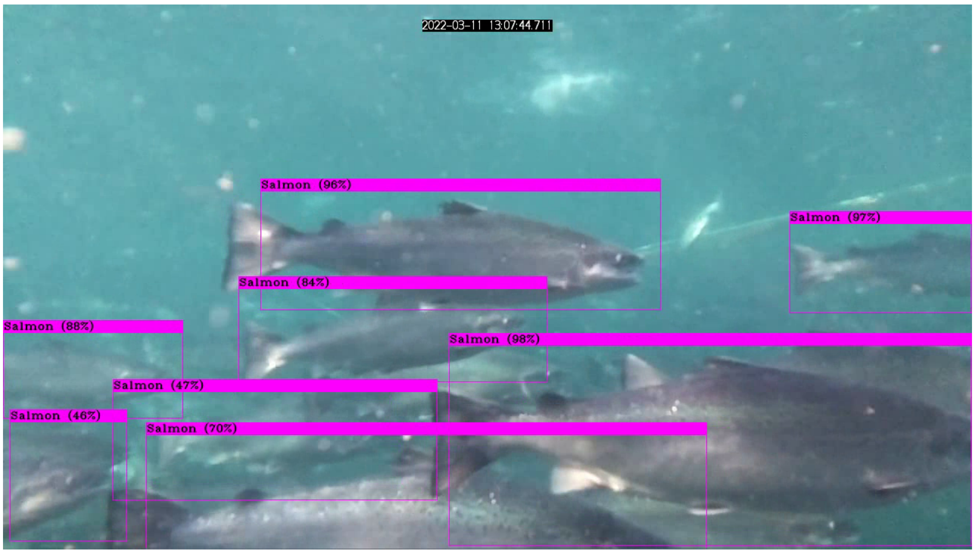
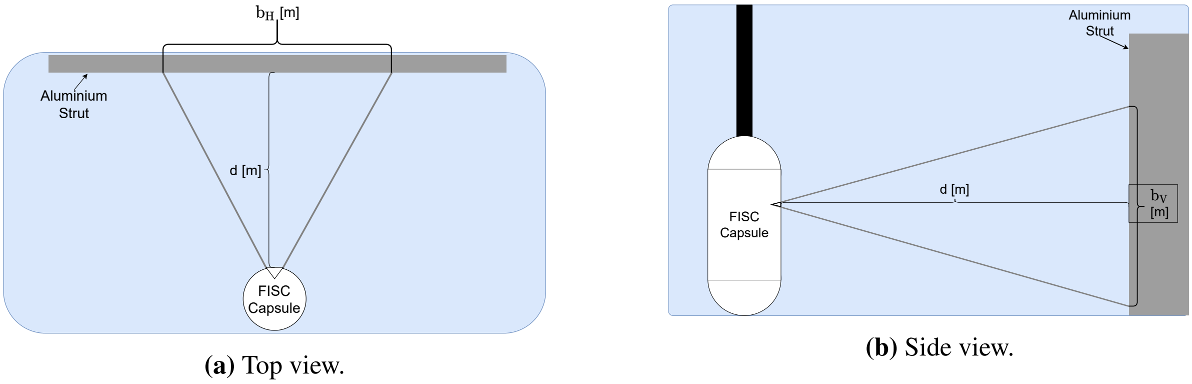


Figure 4.17: Verification of custom YOLOv4 detector functionality.

4.5 FISC Camera Characterization

To enable fish size estimations from underwater images in the post-processing, the horizontal and vertical FOV in the integrated FISC camera was characterized. The FISC capsule was placed in a testing aquarium facing the wall, where aluminum struts were mounted on the opposite side. This setup is shown in Figure 4.18.



(a) Top view.

(b) Side view.

Figure 4.18: FISC camera FOV characterization setup illustration.

Then, following the characterization procedure presented in Section 2.3.1, the points on the aluminum struts that were on the outer edge of the horizontal and vertical video frame were marked with tape and the distances between them were manually measured. The results are presented in Table 4.2.

Axis	d [m]	b [m]	FOV (Equation 2.7)
Horizontal	0.51	0.454	≈ 48 °
Vertical	0.51	0.26	≈ 28.6 °

Table 4.2: FISC camera characterization results.

Note that the vertical FOV of 28.6 ° is relatively close to the estimated combined vertical acoustic beam width of 22.5 °(from Figure 4.6). To optimize the matching of optical and acoustic coverage, the optical post-processing pipeline excludes YOLOv4 detections of Atlantic salmon which have their bounding box center vertical coordinate above or below a vertical threshold. This threshold is depicted as a black line which is visible in all video frames shown in the next section.

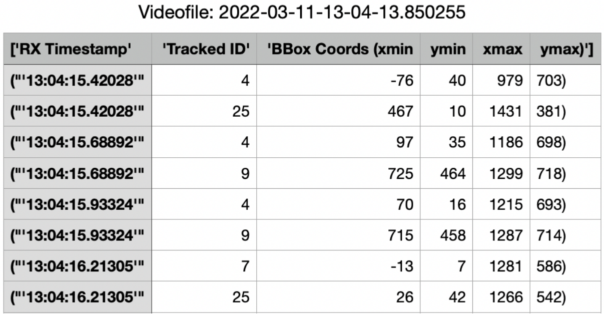
4.6 Optical and Acoustic Sensor Fusion Results

One of the main focuses of the FISC post-processing pipeline has been directed toward utilizing the developed hardware to perform optical and acoustic sensor fusion. The theoretical aspects and implementation procedure have been presented in a step-by-step manner in Chapters 2 and 3 to ensure that the reader can interpret the following presented results. The prerequisite system attributes, i.e. characteristics of the optical and acoustic sensor, have been detailed in the preceding sections, enabling the following sensor fusion.

Only data acquired during the second field test was used for the optical and acoustic sensor fusion post-processing. Most of the presented results below have been hand-picked to illustrate the best-case scenario outputs for the current post-processing implementation.

The *viewSavedData.py* script was used to manually find a selection of videos that seemed to include acoustically trackable salmon by plotting them synchronized in time, as shown previously in Figure 4.12. These selected videos (N=20) were copied to the DeepSORT video directory and passed through the custom *fusion.py*[8] script to generate the tracker data. As detailed in Section 3.2.5, this initiates the modified YOLOv4-DeepSORT software and passes the video and all timestamps of acoustic pings occurring during the video as input. For every video-frame timestamp which coincides with an acoustic ping, all optically tracked Atlantic salmon IDs and their bounding box parameters are added to a temporary array. When the video has ended, the array is exported to a CSV file. A snippet from one of these CSV files is shown in Figure 4.19. Observe that ID # 4 is optically tracked over two subsequent acoustic ping timestamps, which lays the basis for fusion matching between the two detection spaces.

[8] Exact function call: python3 fusion.py --generate



Videofile: 2022-03-11-13-04-13.850255

['RX Timestamp'	'Tracked ID'	'BBox Coords (xmin	ymin	xmax	ymax)']
("'13:04:15.42028'"	4	-76	40	979	703)
("'13:04:15.42028'"	25	467	10	1431	381)
("'13:04:15.68892'"	4	97	35	1186	698)
("'13:04:15.68892'"	9	725	464	1299	718)
("'13:04:15.93324'"	4	70	16	1215	693)
("'13:04:15.93324'"	9	715	458	1287	714)
("'13:04:16.21305'"	7	-13	7	1281	586)
("'13:04:16.21305'"	25	26	42	1266	542)

Figure 4.19: Snippet from generated YOLOv4-DeepSORT CSV tracker-file.

The next step of the post-processing is to utilize these generated CSV files to perform the sensor fusion calculations and time-synchronized plots on tracked individuals (optically) and acoustic pings, one ID at a time. The software reads the CSV file and iterates through it from lowest to highest ID, sorted by their timestamps. Then, the acoustic data for the current timestamp is passed through the CA-CFAR and max peak detection scheme to fetch the range to the (assumed) target in the frame. The echosounder range and ID bounding box center coordinates for the current frame are then passed into a function to calculate the Cartesian coordinates for the object by using Equations 2.9, 2.10 and 2.11 with the characterized camera FOV. The size of the salmon is then estimated by using the bounding box corner coordinates, the estimated true distance to the object ($\mathrm{z}_{\mathrm{coord}}$), and the camera FOV.

The aforementioned values are computed for every acoustic ping and added to a data array. It is therefore possible to estimate the speed of successfully fusion-tracked objects by simply using the Cartesian coordinates in the previous and current time step by using the constant acceleration motion equation ($v = \frac{\Delta d}{\Delta t}$), since the time delta between acoustic pings is known.

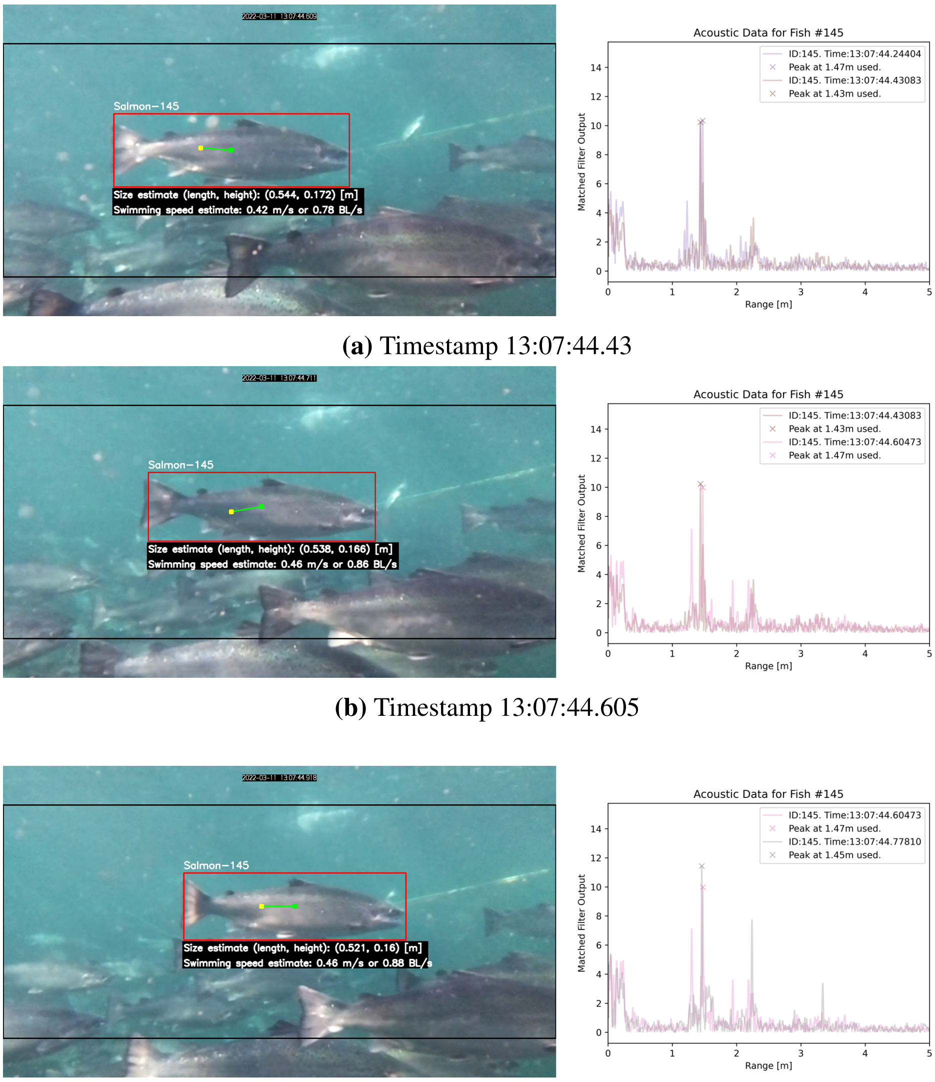
Figure 4.20 shows an exported sequence of the aforementioned processing steps over 3 time steps for one selected individual[9]. The yellow and green dots connected with a line added to the visual output illustrates the bounding box center position change between the previous and current moment in time. The calculated size and velocity of the tracked object is shown on the visual plot, where the velocity is given as an absolute value and BLs$^{-}1$ estimate. This individual was estimated to be ≈0.53 m long and swimming at a velocity of ≈0.45 m/s if the mean values across detections are used.

It is observed in the acoustic plots on the right that the maximum peak at ≈1.45 m is used as the assumed radial distance to the object, where the largest peak dynamics between pings are preserved.

The consecutive size and speed estimates in the three samples are both realistic concerning their

[9] Exact function call: python3 fusion.py --p --ID 145 --start 13:07 --stop 13:08 --savePlots

respective values and very similar to each other, indicating that the chosen peak echo is from this individual.



(a) Timestamp 13:07:44.43

(b) Timestamp 13:07:44.605

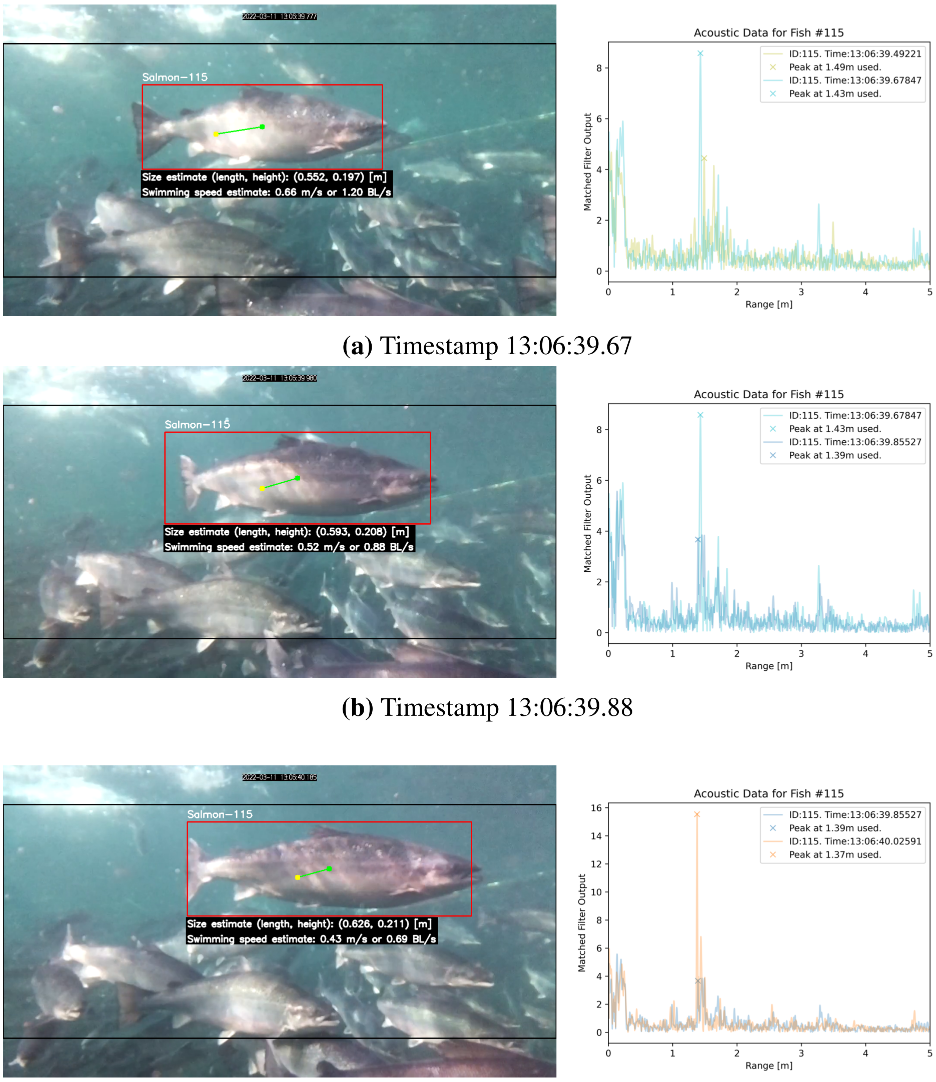
(c) Timestamp 13:07:44.77

Figure 4.20: Fusion output for Salmon #145.

Figure 4.21 shows a similar sequence export of another individual[10], but with a larger variation in estimates of size (Δ_{length}=0.074 m) and swimming speed (Δ_{speed}=0.23 m/s). This individual is estimated to be larger than the previous one, where its mean length is ≈0.59 m and has a mean swimming velocity of ≈0.54 m/s.

The two multi-sequences generated for Salmon #145 and #115 are among some of the few generated outputs where a sequence with a total of four consecutive acoustic pings is successfully matched to the same and presumably correct individual in the sensor fusion software. The software was not able to successfully generate sequences with more than four consecutive optically

[10] Exact function call: python3 fusion.py --p --ID 115 --start 13:06 --stop 13:08 --savePlots



(a) Timestamp 13:06:39.67

(b) Timestamp 13:06:39.88

(c) Timestamp 13:06:40.08

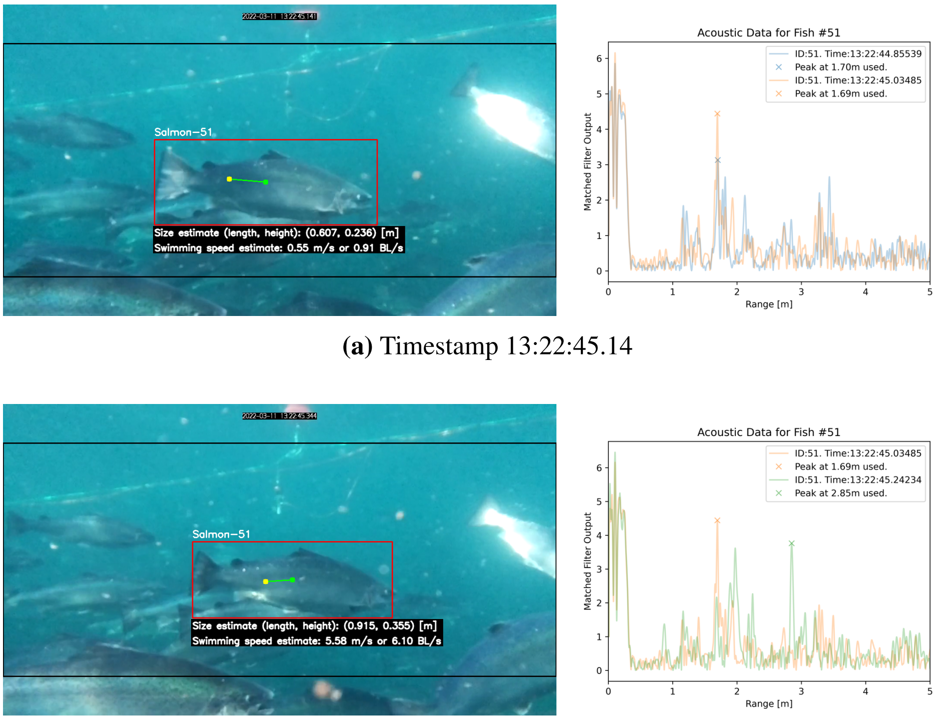
Figure 4.21: Fusion output for Salmon # 115.

and acoustically matched outputs on the available data. Either the YOLOv4-DeepSORT tracker permanently or temporarily lost track of the ID in focus, or the acoustic detection scheme chose an echo that didn't coincide with the correct individual in the video frame as the distance to the target.

An occurrence of the latter case is shown in Figure 4.22[11]. A successful match is shown in Figure 4.22a, but in the next time-step (Figure 4.22b), an acoustic detection 1.16 m further away from the previous target is chosen by the detector. In the first plot, the length (0.6 m) and swimming speed (0.55 m/s) for the individual is realistic in magnitude, and hence likely a correct match. When fusion mismatch occurs in the next acoustic ping, the length and swimming speed are estimated to be 0.9 m and 5.58 m/s, respectively. Fusion mismatch caused by incorrect

[11] Exact function call: python3 fusion.py --p --ID 51 --start 13:22 --stop 13:23 --savePlots

acoustic detection corrupts the size estimates since the assumed distance to the object is wrong. The swimming speed estimates are consequently severely overestimated since the calculated 3D Cartesian coordinates for the object have changed excessively in the depth axis (z).



(a) Timestamp 13:22:45.14

(b) Timestamp 13:22:45.34

Figure 4.22: Fusion output for Salmon # 51.

Additional sequences with two or three successful tracks and realistic estimations were generated from the second field test data, but are not presented since the sensor fusion functionality is demonstrated well enough with the results shown.

Chapter 5

Discussion

5.1 Vertical Acoustic Coverage

In Figure 4.6, the combined vertical beam pattern for the FISC hydroacoustics was presented. This test was performed with a procedure that, theoretically, should produce correct results. However, the resulting beam pattern raises concerns regarding the validity of the characterization. The first sidelobes appeared at approximately 4-5 dB below the peak of the main lobe. This contradicts the principles presented in Section 2.1.1, regarding windowing. Since the rectangular window has the first sidelobe levels at 13 dB below the main lobe peak, this is an expected minimum for this characterization. The raw data from the test looks quite different from the presented result and was filtered to produce the plot shown in Section 4.1. Figure 5.1 below shows the raw data plotted by angle with dB in the Y-axis, before any processing.

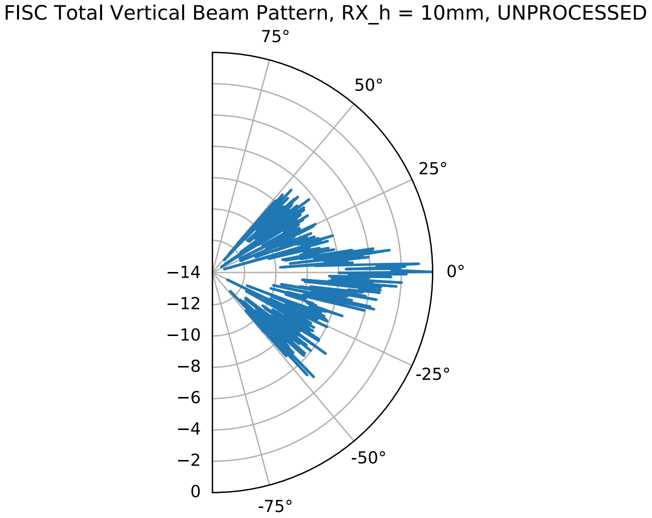


Figure 5.1: Unprocessed FISC vertical beam pattern.

Hence, the raw data indicates that the measurements themselves are suspicious and some additional analysis must be done. A customization to the beam pattern simulation script was used to visualize the theoretical combined beam pattern. Figure 5.2 shows the resulting simulation, where the RX and TX beams are separated by their true Cartesian distance (0.3 m) before summation to yield the combined beam. Figure 5.3 shows the theoretical combined beam in its own plot.

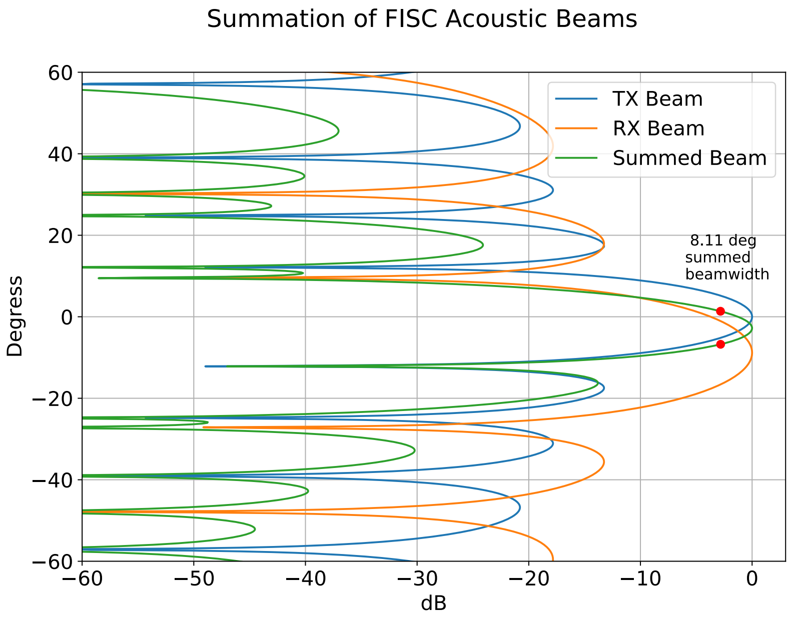


Figure 5.2: Acoustic beam summation simulation.

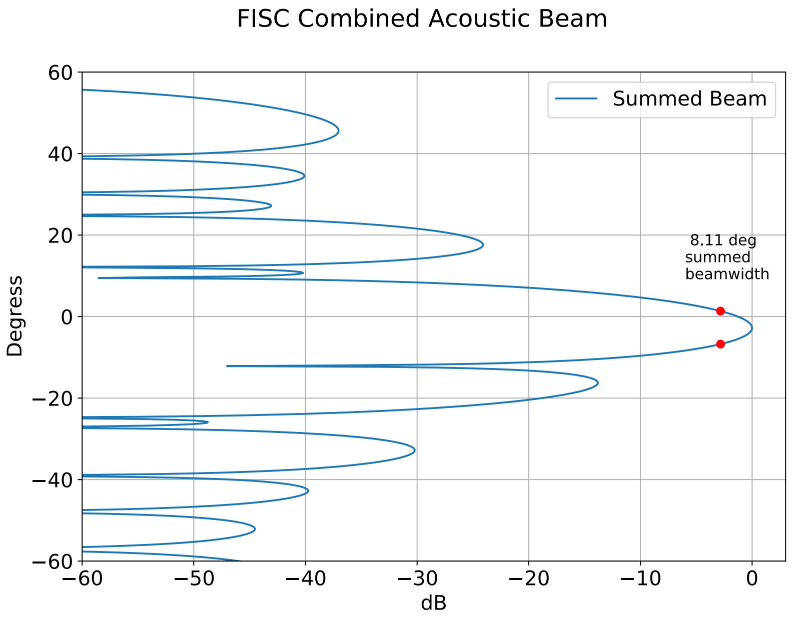


Figure 5.3: Simulated FISC combined acoustic beam.

As observed, the simulated combined beam has an approximated beamwidth of 8.11 °, which is much narrower than the characterization result. Due to time limitations, a new characterization test with the system could not be performed. The noisy unfiltered data is likely caused by poor acoustic detections of the steel ball and, therefore, varying magnitude in the matched filter output in subsequent pings. Moreover, the steel ball was too small to be detected without the matched filter and a larger target was not available at the time.

Although these specific results do not affect the ultimate goals of the thesis, it is important to note that a more precise determination of the vertical acoustic characteristics of the system is desirable in future work. A proposed simple procedure is to use the same installation, but with two modifications that should improve the characterization:

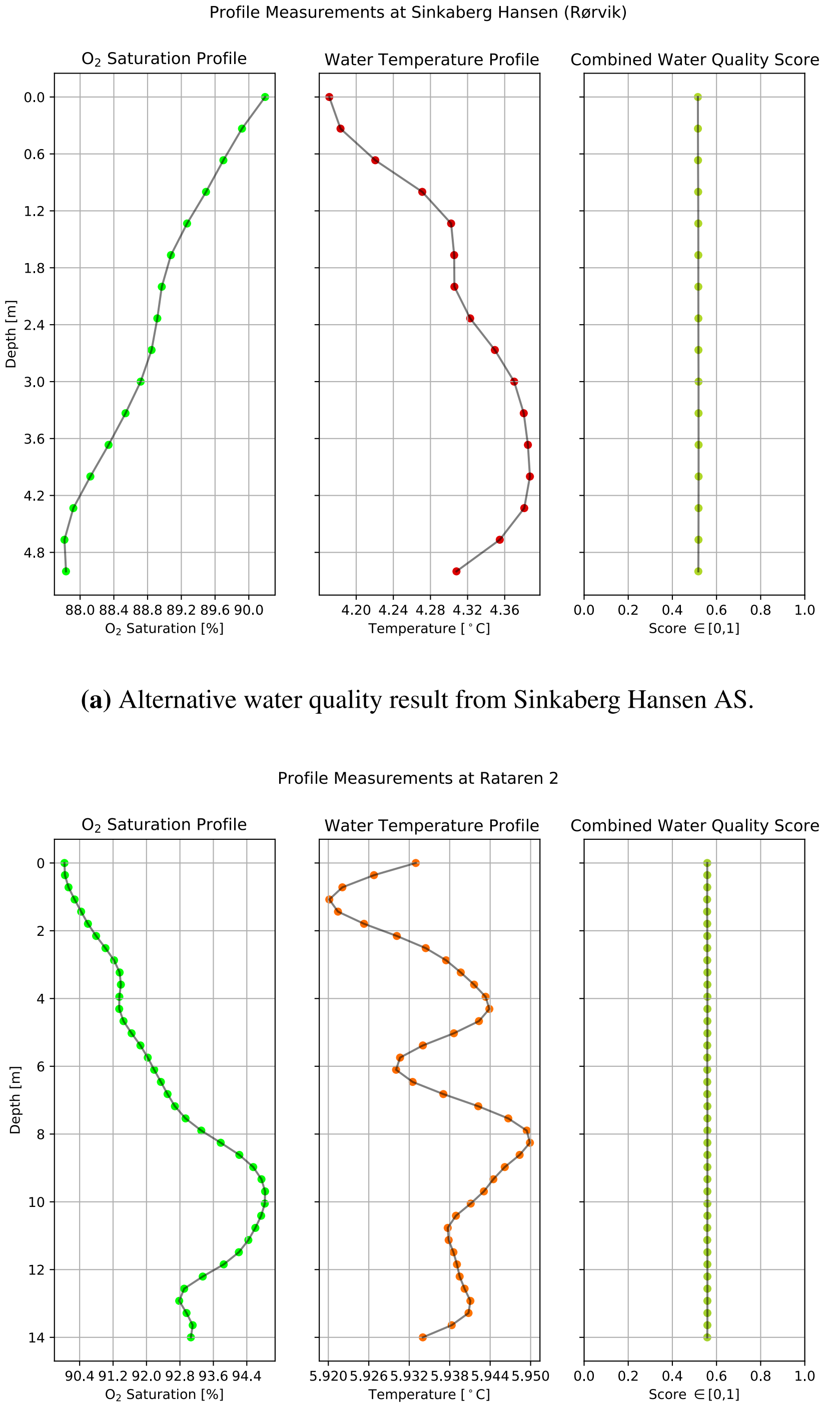
- Use a larger target for improved detections (without matched filter).
- Use a stepper motor to lower the target at a slower, more controlled, and constant speed.

5.2 Combined Water Quality Score

In Sections 4.2 and 4.3, depth profiles of the O_2 and water temperature within the fish-pens during the field tests were presented. In both field tests, the O_2 saturation was within the optimal thresholds for Atlantic salmon ($>$75 %), as detailed in Section 2.5.2. This is also indicated from the green dots in their respective plots (Figure 4.9 and 4.13) generated by the defined quality score functions for these two parameters (Section 3.2.4). The water temperature was however far below the optimum at all depths during both field tests ($\approx$4.28 °C in Rørvik and $\approx$5.9 °C at Rataren II). According to Falconer et al. (2020), temperatures this low result in reduced welfare and food intake, slow growth, increased stress and mortality for Atlantic salmon. Based on the successfully fused optical and acoustic data, the fish exhibited normal behavior (swimming speeds). An apparent fact is that the ocean temperatures vary seasonally, and it is generally expected that the temperatures along the Norwegian coastline are 4-8 °C during the winter season. Nevertheless, these temperatures are not ideal for Atlantic salmon.

Since the water temperature quality score function was defined as a bell curve, which gives a very low score when far from the optimum, the resulting combined water quality score is also very low since the total is a multiplication between the two separate scores.

In the initial field test, the mean combined water quality score over all depth increments was close to zero ($\approx$0.04) due to the low temperatures. At Rataren II, the mean combined score was a bit higher ($\approx$0.12), but still far from the adequate threshold range given by the aquatic environment score heat map (Figure 3.8). One could argue that the multiplication of the two environmental factors is an excessively strict requirement since even if one parameter is well beyond the optimal thresholds, the other parameter will heavily reduce the total quality score if its individual score is low. Changing this requirement from multiplication to e.g. a mean value of the two individual quality score functions, yields the combined water quality scores shown in Figure 5.4.



(a) Alternative water quality result from Sinkaberg Hansen AS.

(b) Alternative water quality result from SINTEF ACE.

Figure 5.4: Water quality depth profiles with alternative combined score requirement.

However, the heat map with this alternative requirement becomes very tolerant to accepting both low water temperatures and low O_2 saturations as adequate values, shown in Figure 5.5. It is still very important to accentuate the fact that the connection between these two environmental parameters and the intricate biological factors resulting in good health and welfare are far more complex than what is assumed in the thesis' implementation. Additionally, an experienced fish health biologist should be included in defining how these separate parameters should be combined and weighted. The approach applied in this work is based on optimal threshold values from cited work and the author's limited knowledge within biology.

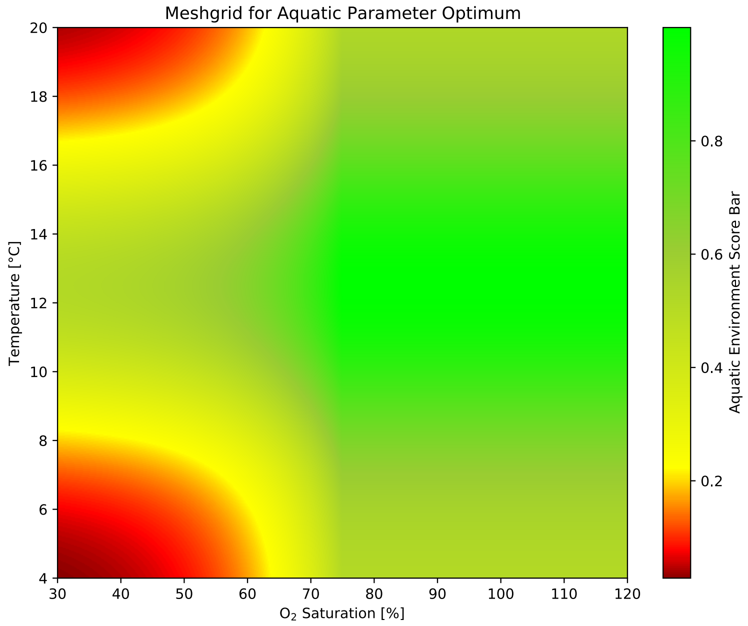


Figure 5.5: Alternative aquatic environment score heat map.

5.3 Water Current Source Direction Estimations

The orientation sensor data from both field tests were used in the post-processing software to estimate from which compass direction water currents were coming from, by using the calculations presented in Section 3.2.3.1.

Figure 4.11, showing the water current direction estimates for the initial field test showed a repeating cycle of decreasing compass heading water current direction. Sadly, no ground truth data on the actual water current directions during this test could be acquired, but both the author and the other participants remember noticing that currents were coming from north-east (45 °) or west (270 °) the two days we were on site. These currents were strong enough to visibly alter the waiting-pen net near the water surface. Unfortunately, extensive analysis of the orientation sensor data from this test gave no results indicating what we observed visually. These results are therefore deemed inconclusive in regard to water current direction estimation.

Figure 4.14 showing the same calculations from the second field test data were comparatively less erratic but still showed some strange alternations in the estimated current source direction. These alternating water current source directions were (approximately) east and south-southwest. From 10:58 until 12:38, excluding the ≈90 minutes between 11:03 and 12:31 when the sensor wasn't sampled at all, the mean estimate indicated that the water currents were coming from the east (≈100 °). Following this, the direction estimate alternated between east and south-southwest throughout the test duration.

Christer Johansen, an operations manager at SalMar ASA, was contacted after the second field test to get information about the true water currents during the data acquisition. He informed that the day we were there, March 11, was the day with the lowest water current strengths that entire month. There was a flood tide during the acquisition period that day (11:00-13:00), meaning that these low currents were coming from the southwest. The presented estimates show alternations where the south-southwest direction could have indicated correct functionality during these periods, but checking the saved videos from these periods revealed that it is more likely that these estimates were caused by influence from the swimming Atlantic salmon. The capsule was located on the east-southeast side of the pen, and the Atlantic salmon were mostly swimming in an anti-clockwise direction during the test. This means that fish colliding with the capsule, or changes in local water currents caused by the fish swimming in large clusters, would generally bump the capsule into orientations indicating a south-southwest water current direction. Figure 5.6 shows a video frame from a situation where this is a likely cause. The fish pen edge is on the right side of the image, meaning that the fish in the frame are swimming north-northeast.

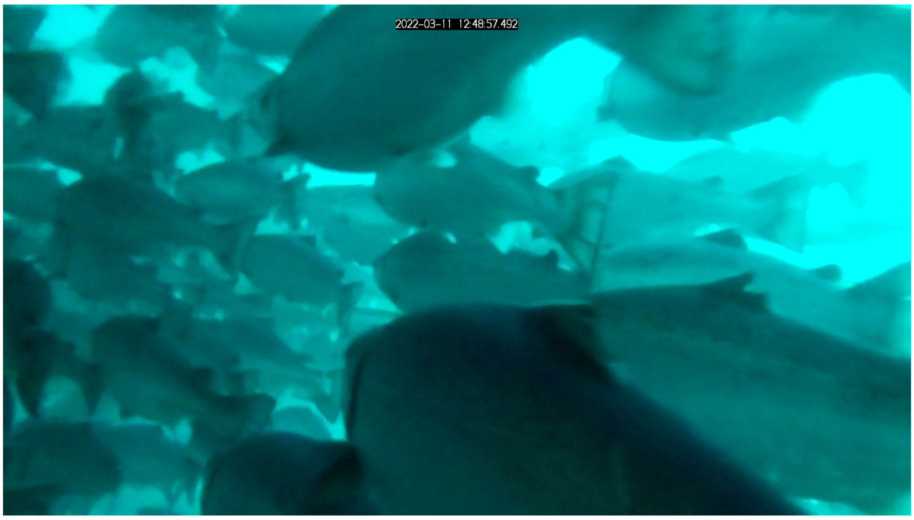


Figure 5.6: Video frame of fish cluster from second field test.

The FISC system is originally intended to be placed in the radial center of fish pens, but such an installation was not possible in either performed field tests. How a center placement would affect the directional vertical inclination estimates is therefore unknown, but there are generally far fewer fish in the center when compared to outer radial locations within the pen. If this decrease in direct physical disturbances would yield correct estimates with the FISC prototype is uncertain. Furthermore, the weight of the FISC capsule is 3.1 kg underwater (4.7 kg in air). Hence, the angle of inclination magnitudes caused by water currents will be small. In possible future versions of the system, the capsule itself would weigh less and several capsules would be integrated on the same cable. This should generally improve the water current estimates since data from orientation sensors at different depths can be combined. Additionally, with a lower weight, the angle of inclination magnitude will increase which ultimately results in more certainty in the estimation.

The only definite fact regarding the implemented directional vertical inclination estimates is that they have been tested and confirmed to be correct in a test tank facility without external influences from e.g. swimming fish. This method of estimating water current direction is hence assumed to be valid and achievable. The physical placement of the FISC capsule during both field tests is concluded to corrupt the presented directional water current estimates, and more tests must be performed to assess the functionality in a full-scale environment.

5.4 Camera Lens Pincushion Distortion

The horizontal and vertical FOV for the integrated FISC camera was detailed in Section 4.5. In the presentation of the theoretical aspects regarding monocular camera geometry (Section 2.3.1), a short remark was made on neglecting the distortions caused by the flat water-tightening lens. These distortions were very apparent during the camera characterization. A recreation of the calibration setup frame is shown in Figure 5.7, where it is easy to observe that the straight aluminum struts on the frame perimeter seem to bend towards the center.

Although the distortions are visibly large towards the frame corners, it is unknown how much the pincushion distortion, in general, affects objects which are closer to the frame center. In the acoustic and visual sensor fusion results, the objects were however not in the outer edges of the frame, and consequences from optical distortions are *assumed* to be negligible.

Shortis (2015) reviews several advanced underwater camera calibration techniques and emphasizes the importance of proper camera calibration to ensure precise and reliable size measurements, especially in the case of fish biomass estimation. The importance of lens shape (flat or domed) is also addressed, where domed lenses can counteract the pincushion distortion caused by submerging optical sensors.

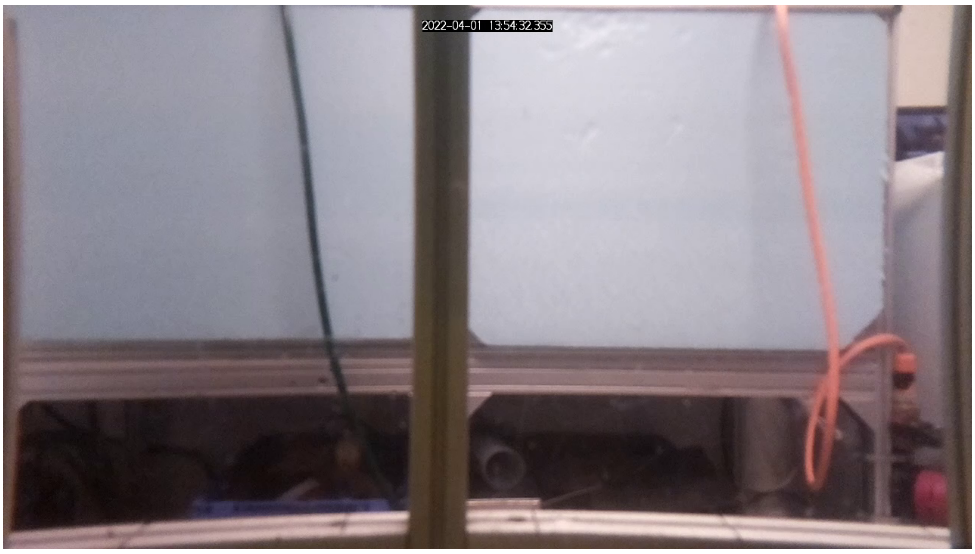


Figure 5.7: Example of pincushion distortion from FISC camera frame.

Other intrinsic calibration procedures than the applied approach were not considered due to a lack of time and limited knowledge within the field of optics. The most important point to be made is that optical systems designated for underwater use require proper design and calibration routines to ensure high accuracy. In this project, however, a low-cost camera with a flat plastic lens and rudimentary FOV characterization underwater was used. The magnitude of how this simplified approach affects size estimate errors is unknown but should be addressed if further development will take place.

5.5 Feasibility of Optical and Acoustic Sensor Fusion

The desired goal for the optical and acoustic sensor fusion in the thesis was to successfully fuse these data to yield individual size and swimming speed estimates of Atlantic salmon. In Section 4.6, a selection of presented results showed that this goal was indeed achieved, with realistic estimates.

An optical and acoustic sensor fusion approach similar to the one investigated in this thesis was implemented by Roznere et al. (2020). They used a monocular camera and a single-beam echosounder, both mounted on an underwater Remotely Operated Vehicle (ROV), to improve depth ambiguity in their Simultaneous Localization and Mapping (SLAM) operations. Specifically, they used ORB-SLAM2, which is commonly applied in monocular SLAM operations to yield sparse 3D reconstruction and estimate real-time camera trajectory[1]. Although their fusion procedure is directed toward correcting depth ambiguity, the fusion ideology and physical installation are quite similar to the one applied in this thesis. They performed a calibration

[1] `https://github.com/raulmur/ORB_SLAM2`

to minimize the localization errors caused by the installation distance between the two sensors, i.e. extrinsic calibration. Their results show solid depth ambiguity improvements during their SLAM operations, indicating that echosounder and camera sensor fusion has much potential.

In this thesis, no optical-acoustic extrinsic calibration was performed. Hence, the possible improvements this calibration would have on the presented results are unknown. Ferreira et al. (2016) emphasizes that extrinsic calibration is important to improve the performance of this type of sensor fusion, but is not trivial to carry out. Time limitations restricted the possibilities of attempting extrinsic calibration during this project. Either way, there are additional aspects surrounding the implemented optical and acoustic fusion processing that must be discussed to better determine the feasibility of a system utilizing these sensors for the applied purposes.

5.5.1 Bounding Box Accuracy

In all presented results of the optical and acoustic sensor fusion, it is easy to observe that the bounding boxes around the Atlantic salmon do not encapsulate the individuals perfectly. This obviously affects the individual size estimates if the bounding box size is wrong. In some of the presented results, the bounding box size and center accuracy also vary for the same individual during the sequence.

The main reason behind the bounding box inaccuracies is, with a high level of certainty, caused by the low amount and low variety of data used for training the implemented YOLOv4 model. The custom YOLOv4 detector training was detailed in Section 4.4. As was mentioned here, 184 labeled images of Atlantic salmon and 184 unlabeled underwater images with other objects were used in the implemented model. A general rule of thumb for training deep neural networks is to include a minimum of 500 training samples per class. The two last YOLO model training routines (see Appendix G) used the mAP parameter during training. It is observed that the mAP in both these runs is 99.5 % towards the end of training. This can be an indicator of overfitting to the low amount of training data, but this is ultimately an analysis that is out of the scope of this thesis.

Zhu et al. (2015) details various aspects when it comes to the quality and amount of training data for Deep learning. This paper mainly focuses on Deep learning theory which is somewhat unrelated to this case and is beyond the scope of this thesis. Nevertheless, it is presented that the average number of training images per class correlates with the trained model's average precision when looking at the top performers in the PASCAL VOC dataset between 2006 and 2011. They detail that both the amount and quality of training data are important to achieve optimal performance in visual object detection.

Similar remarks are made by Lei et al. (2019), where how the accuracy of object detectors using Deep learning is affected by the number of training images (N). Their results indicate that increasing the amount of training data results in a more accurate model but only to a certain point. This point (N>5000) is however far beyond the number of training images used in the implemented model (N=368), illustrating the deficiency of training data in this project.

Lastly, the following remark is made in the official YOLOv4 release repository[2]: *"you should preferably have 2000 different images for each class or more, and you should train 2000*classes iterations or more"*.

The implemented YOLOv4 model accuracy can therefore be assumed to have a large room for accuracy improvement if much more training data is acquired. Nevertheless, the implemented model shows an accuracy that arguably performs decently given the low-cost camera and very small amount of training data.

It is also important to mention that the side-view length of fish varies as they swim due to how they curve their bodies to induce forward motion. This is a fact that is neglected throughout the thesis but should be mentioned as a side-note to the above size-estimation discussion.

5.5.2 Automatic Target Matching

As mentioned throughout the post-processing design and presentation of the results, the acoustic detection scheme simply uses the largest CA-CFAR detector output as the assumed acoustic target in focus during the fusion routine. This in itself heavily restricts the amount of possible successful fusion matches to those sequences where one individual is a certain distance from others and in relatively close proximity to the FISC capsule. If several targets are within the acoustic beam and they return echoes that alternate in strength over a few pings, the acoustic detection scheme will likely fail to detect the same and/or correct individual. This situation was presented in Figure 4.22b.

In essence, an acoustic peak tracking algorithm should be implemented to avoid these occurrences. Due to a lack of time, the relatively simple approach with max peak detection and no tracking in the acoustic data was used since a selection of desirable results were achieved regardless. A very quick fix to this specific issue would be simply to accept detections that are maximum N range cells away from the previously selected detection. This would however only solve half of the root problem since it is not certain that the initially chosen acoustic detection is the one matching the optically tracked target in focus.

Increasing the acoustic ping rate is also a simple approach to decrease the amount of lost acoustic tracks since this would decrease the changes in dynamics between every ping. The system

[2] `https://github.com/AlexeyAB/darknet`

in its current state was not capable of going higher than 6 Hz, mostly due to inefficiency in the Python acquisition software.

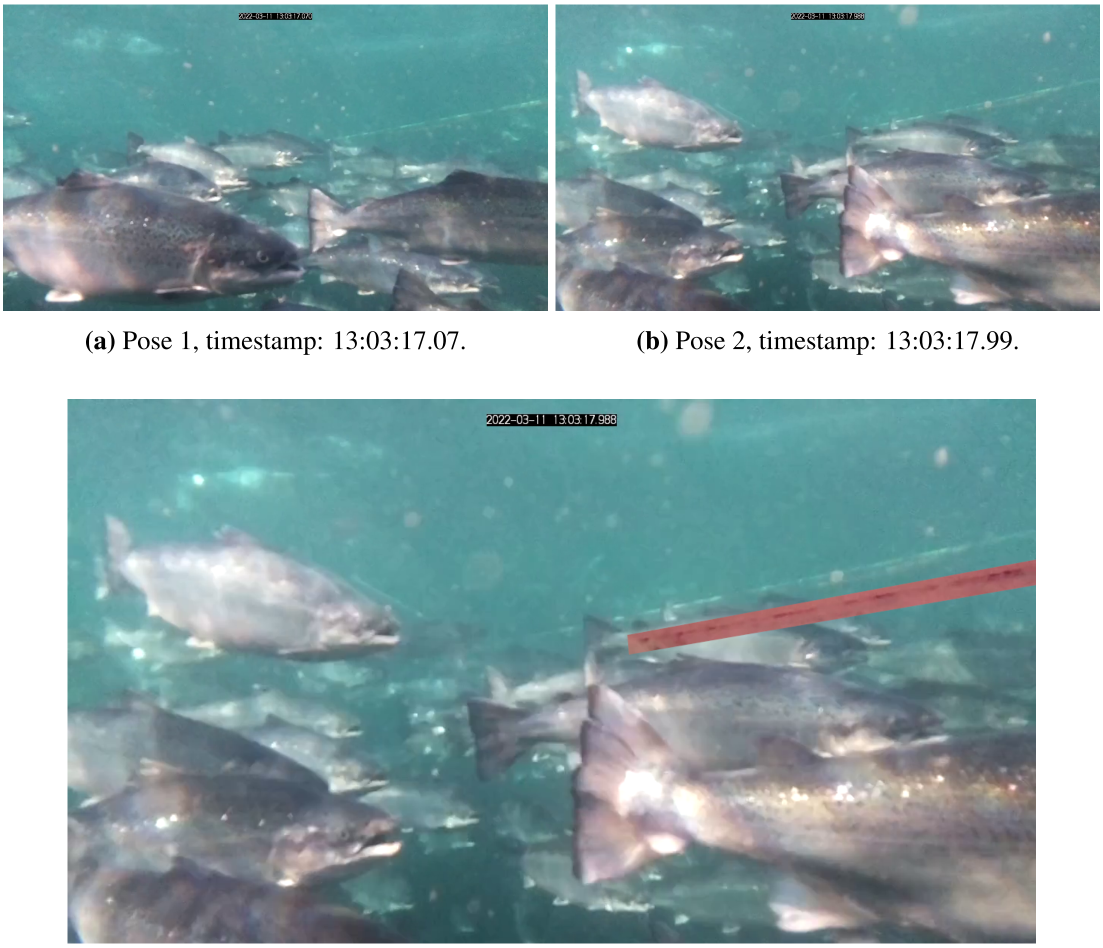
Ultimately, the author believes the best approach would be to develop an algorithm that assigns an acoustic detection to its matched bounding box detection based on applying *a priori* information. A proposal here would be to generate probability statistics based on the current bounding box scale in pixels and all CA-CFAR acoustic detections to determine which detection is most likely to match the current bounding box. This should be possible since prior information on the approximate expected size of the Atlantic salmon is available based on age and manual size measurements which are performed regularly in aquaculture. With this information, the object size can be estimated with all CA-CFAR acoustic detections to choose the detection range which gives the most likely true target scale. Although this approach could fix cases where the current FISC post-processing chooses a detection that is obviously at the wrong range, it is difficult to foresee how it would perform when several acoustic detections are separated by only a few range cells and the estimated sizes are very similar.

The aforementioned approach would also enable more efficient parallel processing of the data. In its current state, the post-processing software analyzes one video at a time and iterates over each ID tracked in the video. If a total of 100 individuals are tracked by the YOLOv4-DeepSORT implementation during one video, and N is the number of acoustic pings during the video, the post-processing software will analyze the same generated CSV file 100*N times and extract video frames coinciding with every ping. With a ping rate of 6 Hz and 15 s videos, N equals 90, and the CSV file is hence analyzed 9000 times in the worst-case scenario. This assumes that the same individuals are tracked during the entirety of the video, which never occurred in the field tests, but still illustrates the inefficient structure of the software in its current state. If the aforementioned automatic target matching approach is introduced, it would be possible to pass all YOLOv4-DeepSORT detections in the current timestamp through an algorithm to match all current CA-CFAR peak detections to their respective bounding boxes. Intuitively, this approach seems feasible and effective, but essentially requires an extensive reconstruction of the post-processing software.

5.5.3 Importance of Camera Pose

To maximize the acoustic ping rate, the orientation sensor was not sampled between pings during the second field test, as mentioned in Section 4.3. At that time, this choice was assumed to have minimal consequences for the post-processing. However, due to the FISC capsule hanging freely from the interface cable with many Atlantic salmon swimming near the cable/capsule in the water column, capsule motion and rotation were observed in many of the recorded videos. This motion gives an error in the change of 3D Cartesian coordinate estimates for fused detection matches since the current processing software assumes that the FISC capsule has a static pose during the fusion processing sequence.

Figure 5.8 shows an example case of pose-change during ≈1 s in a randomly selected video from the second field test. Figure 5.8c shows a merged image where the red cutout shows how far the stationary pen perimeter tube moves in the frame during 1 s.



(a) Pose 1, timestamp: 13:03:17.07.

(b) Pose 2, timestamp: 13:03:17.99.

(c) Merged visualization of pose change, illustrated with red bar.

Figure 5.8: FISC camera pose change over 1 s.

Since the orientation sensor is not sampled in these sequences, compensation for the change in pose was not possible to implement. How much this affects estimates of swimming speed is directly dependent on how much the camera pose changes between pings in these sequences. The results presented in Section 4.6 were however from optical sequences with minuscule changes in camera pose.

The induced motion and rotation are in itself not a major issue and is to be expected in this type of system deployment scheme. However, it is important to know the change in pose for the sensor fusion estimates to become more accurate. This could have been solved in the current system prototype by modifying the FISC capsule Raspberry Pi to directly sample the orientation sensor and store this data locally instead of immediately relaying it to the topside PC, which would not affect the acoustic ping rate.

5.5.4 Real-time Processing

The integrated FISC topside PC has not generated any of the post-processing-related results in the thesis. However, it is fully capable of running all post-processing software, but not as quickly when compared to computers with more processing power. Since the development of the post-processing software was done on a relatively powerful computer with a dedicated GPU, the actual processing and exporting of results were achieved on the same computer, enabling a lot of trial and error without substantial computation time, which was an important factor during development.

Since the post-processing software (available at (Ericsson, 2022) includes pre-generated DeepSORT tracker CSV files, any average modern computer should be able to reproduce the presented results quite rapidly. It is only the *generation* of the CSV tracker files that rely on a GPU for fast computation since this process runs the implemented YOLOv4 object detector and DeepSORT MOT software. Visualizing the tracker data, which is what the sensor fusion processing mostly does, only relies on loading the video files and extracting a single frame at every acoustic ping. This is rapidly generated on a modern computer or laptop, including the current FISC topside PC.

The inclusion of a dedicated GPU in the topside cabinet would enable close to real-time processing by generating tracker data on newly recorded videos. This is simply a short remark since the prototype is at a stage where real-time processing is unnecessary.

Chapter 6

Concluding Remarks

The fish-farming industry has a desire to gain more control during daily operations to maximize profits and minimize health-related issues for the biomass they produce. Given the large aquatic volume of the fish pen and the complexity of fish-specific biological factors, better control is not easy to attain. A starting point to achieve more control is to utilize systems that can reliably extract accurate and, most importantly, valuable data. These data should ideally be on the aquatic environment and the fish themselves since data from one without the other rarely gives conclusive answers to issues that may arise.

Several products on the market have been developed to target specific known issues in the industry, such as lice-counting, accurate size estimation, and general welfare determination. Most of these products are highly technologically advanced, and therefore expensive.

The hardware and initial software development of the FISC prototype began approximately 7 months before the master's project. The main objective for FISC was to be a multi-sensor proof of concept for a future product targeted toward the fish-farming industry. Its purpose was to investigate if data from acoustic, optical, orientation, and environmental sensors enclosed in the same mechanical housing could be feasible for the aquaculture industry. Specifically, a basic fusion of two environmental parameters (O_2 and temperature) as well as an optical and acoustic sensor fusion approach was investigated. The thesis itself has mainly focused on the development and testing of the post-processing and fusion-related software, which uses data acquired with the FISC prototype from two full-scale field tests.

Orientation data was used to estimate from which direction water currents were coming by decomposing the FISC capsule's absolute orientation vectors. The ideology behind this calculation was that forces from water currents would result in an inclination caused by them, but the full-scale results were inconclusive due to erratic estimates. The main cause was assumed to be from physical influence from the fish since the capsule was lowered in a location in the direct swimming path of the Atlantic salmon.

A self-defined approach was used to compute a water quality score from the field tests by using the waters O_2 saturation and temperature with their respective optimal thresholds for Atlantic salmon (from prior research). The results based on the implemented approach showed an optimal O_2 saturation, but water temperatures were well below the optimum. The result in the combined water quality score proved to be quite low. In the discussion of these results, an alternative and less strict quality score procedure was introduced, but the combined scoring ultimately requires experts in fish biology to determine the best approach.

The application of Computer Vision through Deep learning laid the basis for the optical and acoustic sensor fusion approach. A custom YOLOv4 object detector was trained on manually labeled data to detect Atlantic salmon, and this model was integrated with the post-processing software through an open-sourced implementation of the DeepSORT multiple object tracker. Necessary modifications to the DeepSORT software were performed to extract tracked fish IDs and their pixel positions in every frame that coincides with an acoustic ping timestamp. These data, together with the intrinsic characterization of the integrated FISC camera and hydroacoustic sensors, allowed for 3D Cartesian coordinate determination of Atlantic salmon. These generated 3D coordinates were then used to estimate the size and swimming speed of fusion-matched individuals. The presented results showed that the developed software was successful at estimating the size and swimming speed of individual fish, although with some small variations in accuracy and several occurrences of fusion mismatch. Conclusively, the implemented optical and acoustic sensor fusion relies on further development to become a viable solution for a future product on market. The specific issues to target in this domain were detailed in the discussion.

The initially set goals for the thesis, and project in general, were met. The developed electronic and acoustic hardware functioned as intended. The system is capable of acquiring data over long periods of time, both in a test facility and in its true environment. The developed post-processing software is able to generate various representations of the acquired data. These functionalities include but are not limited to, depth-profiles of environmental parameters with water quality score determination, running YOLOv4-DeepSORT to generate tracker files as well as visualizing time-synchronized optical and acoustic sensor fusion. What lies ahead for the FISC system is currently unknown, but hopefully further development based on the findings in this thesis can bring this multi-sensor solution to an industrialization stage in the future.

Bibliography

Abraham, D.A. (2017). "Chapter 11 - Signal Processing". In: *Applied Underwater Acoustics*. Ed. by Thomas H. Neighbors and David Bradley. Elsevier, pp. 743–807. ISBN: 978-0-12-811240-3. DOI: https://doi.org/10.1016/B978-0-12-811240-3.00011-4. URL: https://www.sciencedirect.com/science/article/pii/B9780128112403000114.

Alexey et al. (Oct. 2021). *AlexeyAB/darknet: YOLOv4*. Version yolov4. DOI: 10.5281/zenodo.5622675. URL: https://doi.org/10.5281/zenodo.5622675.

Bailey, Jennifer L. and Sigrid Sandve Eggereide (2020). "Indicating sustainable salmon farming: The case of the new Norwegian aquaculture management scheme". In: *Marine Policy* 117, p. 103925. ISSN: 0308-597X. DOI: https://doi.org/10.1016/j.marpol.2020.103925. URL: https://www.sciencedirect.com/science/article/pii/S0308597X1930452X.

Bewley, Alex, ZongYuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft (2016). "Simple Online and Realtime Tracking". In: *CoRR* abs/1602.00763. arXiv: 1602.00763. URL: http://arxiv.org/abs/1602.00763.

Bjelland, Hans V. et al. (2015). "Exposed Aquaculture in Norway". In: *OCEANS 2015 - MTS/IEEE Washington*, pp. 1–10. DOI: 10.23919/OCEANS.2015.7404486.

Blanco, Jose Luis (Sept. 2010). "A tutorial on SE(3) transformation parameterizations and on-manifold optimization". In.

BNO055 Datasheet (2020). Rev. 1.7. Bosch Sensortec. URL: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bno055-ds000.pdf.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *CoRR* abs/2004.10934. arXiv: 2004.10934. URL: https://arxiv.org/abs/2004.10934.

Brigham, E. Oran (1988). *The Fast Fourier Transform and Its Applications*. USA: Prentice-Hall, Inc. ISBN: 0133075052.

Burt, Kim, Dounia Hamoutene, Gehan Mabrouk, Chris Lang, Thomas Puestow, Dwight Drover, Randy Losier, and Fred Page (2012). “Environmental conditions and occurrence of hypoxia within production cages of Atlantic salmon on the south coast of Newfoundland”. In: *Aquaculture Research* 43.4, pp. 607–620. DOI: https://doi.org/10.1111/j.1365-2109.2011.02867.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-2109.2011.02867.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2109.2011.02867.x.

Chanduri, Sai Shyam, Zeeshan Khan Suri, Igor Vozniak, and Christian Müller (2021). “CamLessMonoDepth: Monocular Depth Estimation with Unknown Camera Parameters”. In: *CoRR* abs/2110.14347. arXiv: 2110.14347. URL: https://arxiv.org/abs/2110.14347.

Chen, Shengyong, Yingkun Xu, Xiaolong Zhou, and Fenfen Li (Jan. 2019). “Deep Learning for Multiple Object Tracking: A Survey”. In: *IET Computer Vision* 13. DOI: 10.1049/iet-cvi.2018.5598.

Ericsson, Martin (2021). *Sensor Cluster for Environmental and Biological Surveillance in Fish Farming Aquaculture*. TFE4590 - Specialization Project Report. NTNU, IES. URL: https://drive.google.com/file/d/1Ca7iIyDDTLzj_y0IET-skUGg9VMeA5tt/view?usp=sharing.

— (2022). *FISC Post Processing and Sensor Fusion Repository*. https://github.com/MartinCKE/FISC-Post-Processing-and-Sensor-Fusion.

Everingham, Mark, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman (June 2010). “The Pascal Visual Object Classes (VOC) Challenge”. In: *Int. J. Comput. Vision* 88.2, pp. 303–338. ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4. URL: https://doi.org/10.1007/s11263-009-0275-4.

Falconer, Lynne, Solfrid Sætre Hjøllo, Trevor C. Telfer, Bruce J. McAdam, Øystein Hermansen, and Elisabeth Ytteborg (2020). “The importance of calibrating climate change projections to local conditions at aquaculture sites”. In: *Aquaculture* 514, p. 734487. ISSN: 0044-8486. DOI: https://doi.org/10.1016/j.aquaculture.2019.734487. URL: http://www.sciencedirect.com/science/article/pii/S0044848619316199.

FAO (2018). *The State of World Fisheries and Aquaculture 2018*. URL: https://www.fao.org/3/i9540en/i9540en.pdf.

Ferreira, Fausto, Diogo Machado, Gabriele Ferri, Samantha Dugelay, and John Potter (2016). “Underwater optical and acoustic imaging: A time for fusion? a brief overview of the state-of-the-art”. In: *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6. DOI: 10.1109/OCEANS.2016.7761354.

Føre, Martin, Kevin Frank, et al. (2018). "Precision fish farming: A new framework to improve production in aquaculture". In: *Biosystems Engineering* 173. Advances in the Engineering of Sensor-based Monitoring and Management Systems for Precision Livestock Farming, pp. 176–193. ISSN: 1537-5110. DOI: `https://doi.org/10.1016/j.biosystemseng.2017.10.014`. URL: `http://www.sciencedirect.com/science/article/pii/S1537511017304488`.

Føre, Martin, Eirik Svendsen, Jo Arve Alfredsen, Ingebrigt Uglem, Nina Bloecher, Harald Sveier, Leif Magne Sunde, and Kevin Frank (2018). "Using acoustic telemetry to monitor the effects of crowding and delousing procedures on farmed Atlantic salmon (Salmo salar)". In: *Aquaculture* 495, pp. 757–765. ISSN: 0044-8486. DOI: `https://doi.org/10.1016/j.aquaculture.2018.06.060`. URL: `https://www.sciencedirect.com/science/article/pii/S0044848617324407`.

Gismervik, Kristine, Brit Tørud, Tore S. Kristiansen, Tonje Osmundsen, Kristine Vedal Størkersen, Christian Medaas, Marianne Elisabeth Lien, and Lars Helge Stien (2020). "Comparison of Norwegian health and welfare regulatory frameworks in salmon and chicken production". In: *Reviews in Aquaculture* 12.4, pp. 2396–2410. DOI: `https://doi.org/10.1111/raq.12440`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1111/raq.12440`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1111/raq.12440`.

Godard, Clément, Oisin Mac Aodha, and Gabriel J. Brostow (2018). "Digging Into Self-Supervised Monocular Depth Estimation". In: *CoRR* abs/1806.01260. arXiv: `1806.01260`. URL: `http://arxiv.org/abs/1806.01260`.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press.

Hanson, Andrew J. (2006). *Visualizing Quaternions*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 9780080474779.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385. arXiv: `1512.03385`. URL: `http://arxiv.org/abs/1512.03385`.

Holm, William A., ed. (2010). *Principles of Modern Radar: Basic principles*. Institution of Engineering and Technology. URL: `https://digital-library.theiet.org/content/books/ra/sbra021e`.

Hovem, Jens M. (2012). *Marine Acoustics The Physics of Sound in Underwater Environments*. Los Altos Hills, California: Peninsula Publishing. ISBN: 9780932146656.

Hvas, Malthe and Frode Oppedal (2019). "Physiological responses of farmed Atlantic salmon and two cohabitant species of cleaner fish to progressive hypoxia". In: *Aquaculture* 512, p. 734353. ISSN: 0044-8486. DOI: `https://doi.org/10.1016/j.aquaculture.`

2019.734353. URL: https://www.sciencedirect.com/science/article/pii/S0044848619313845.

Jaulin, Luc (2015). "7 - Kalman Filter". In: *Mobile Robotics*. Ed. by Luc Jaulin. Elsevier, pp. 219–294. ISBN: 978-1-78548-048-5. DOI: https://doi.org/10.1016/B978-1-78548-048-5.50007-3. URL: https://www.sciencedirect.com/science/article/pii/B9781785480485500073.

Johansson, David, Kari Ruohonen, Anders Kiessling, Frode Oppedal, Jan-Erik Stiansen, Mark Kelly, and Jon-Erik Juell (2006). "Effect of environmental factors on swimming depth preferences of Atlantic salmon (Salmo salar L.) and temporal and spatial variations in oxygen levels in sea cages at a fjord site". In: *Aquaculture* 254.1, pp. 594–605. ISSN: 0044-8486. DOI: https://doi.org/10.1016/j.aquaculture.2005.10.029. URL: https://www.sciencedirect.com/science/article/pii/S0044848605006113.

Jónsdóttir, Kristbjörg, Zsolt Volent, and Jo Arve Alfredsen (Jan. 2021). "Current flow and dissolved oxygen in a full-scale stocked fish-cage with and without lice shielding skirts". In: *Applied Ocean Research* 18, p. 102509. DOI: 10.1016/j.apor.2020.102509.

Kalman, R. E. (Mar. 1960). "A New Approach to Linear Filtering and Prediction Problems". In: *Journal of Basic Engineering* 82.1, pp. 35–45. ISSN: 0021-9223. DOI: 10.1115/1.3662552. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf. URL: https://doi.org/10.1115/1.3662552.

Kok, Manon, Jeroen D. Hol, and Thomas B. Schön (Nov. 2017). "Using Inertial Sensors for Position and Orientation Estimation". In: 11.1–2, pp. 1–153. ISSN: 1932-8346. DOI: 10.1561/2000000094. URL: https://doi.org/10.1561/2000000094.

Kruegle, Herman and Frank Abram (2006). "Chapter 4 - Lenses and Optics". In: *CCTV Surveillance (Second Edition)*. Ed. by Herman Kruegle and Frank Abram. Second Edition. Burlington: Butterworth-Heinemann, pp. 71–107. ISBN: 978-0-7506-7768-4. DOI: https://doi.org/10.1016/B978-075067768-4/50007-1. URL: https://www.sciencedirect.com/science/article/pii/B9780750677684500071.

Lei, Suhua, Huan Zhang, Ke Wang, and Zhendong Su (2019). *How Training Data Affect the Accuracy and Robustness of Neural Networks for Image Classification*. URL: https://openreview.net/forum?id=HklKWhC5F7.

Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *CoRR* abs/1405.0312. arXiv: 1405.0312. URL: http://arxiv.org/abs/1405.0312.

Lovdata (2008). *Forskrift om drift av akvakulturanlegg (akvakulturdriftsforskriften)*. Accessed: 2022-04-26. URL: https://lovdata.no/dokument/SF/forskrift/2008-06-17-822.

Lyons, Richard (Aug. 2011). *Understanding Digital Signal Processing (3rd Edition)*. ISBN: 013702741-9.

Mahafza, Bassem R., Scott C. Winton, and Atef Z. Elsherbeni (2021). *Handbook of Radar Signal Analysis (Advances in Applied Mathematics)*. 1st ed. Chapman and Hall/CRC. ISBN: 9781138062863; 1138062863.

Menna, Fabio, Erica Nocerino, Francesco Fassi, and Fabio Remondino (Jan. 2016). "Geometric and Optic Characterization of a Hemispherical Dome Port for Underwater Photogrammetry". In: *Sensors* 16, p. 48. DOI: 10.3390/s16010048.

Milan, Anton, Laura Leal-Taixé, Ian D. Reid, Stefan Roth, and Konrad Schindler (2016). "MOT16: A Benchmark for Multi-Object Tracking". In: *CoRR* abs/1603.00831. arXiv: 1603.00831. URL: http://arxiv.org/abs/1603.00831.

Moe, Eirik, Merete Skage, and Maria B. Helsengreen (2008). *The Norwegian aquaculture analysis 2021*. Accessed: 2022-05-02. URL: https://go.ey.com/3CXClZi.

Mohinder S. Grewal Lawrence R. Weill, Angus P. Andrews (2007). *Global Positioning Systems, Inertial Navigation, and Integration*. 2nd ed. Wiley-Interscience. ISBN: 9780470041901.

Object Detection on COCO test-dev (2022). https://paperswithcode.com/sota/object-detection-on-coco. Accessed: 2022-03-02.

Oppedal, Frode, Tim Dempster, and Lars H. Stien (2011). "Environmental drivers of Atlantic salmon behaviour in sea-cages: A review". In: *Aquaculture* 311.1, pp. 1–18. ISSN: 0044-8486. DOI: https://doi.org/10.1016/j.aquaculture.2010.11.020. URL: https://www.sciencedirect.com/science/article/pii/S0044848610007933.

Redmon, Joseph (2013–2016). *Darknet: Open Source Neural Networks in C*. http://pjreddie.com/darknet/.

Redmon, Joseph, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi (2015). "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640. arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.

Redmon, Joseph and Ali Farhadi (2016). "YOLO9000: Better, Faster, Stronger". In: *CoRR* abs/1612.08242. arXiv: 1612.08242. URL: http://arxiv.org/abs/1612.08242.

— (2018). "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767. arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767.

Roznere, Monika and Alberto Quattrini Li (2020). "Underwater Monocular Image Depth Estimation using Single-beam Echosounder". In: *2020 IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), pp. 1785–1790. DOI: 10.1109/IROS45743.2020.9340919.

Shkel, Andrei M. (2021). *Pedestrian Inertial Navigation with Self-Contained Aiding (IEEE Press Series on Sensors)*. 1st ed. Wiley-IEEE Press. ISBN: 9781119699552; 111969955X.

Shortis, Mark (Dec. 2015). “Calibration Techniques for Accurate Measurements by Underwater Camera Systems”. In: *Sensors* 15, p. 30810. DOI: 10.3390/s151229831.

Sommerset, Ingunn, Britt Bang Jensen, Geir Bornø, Asle Haukaas, and Edgar Brun (Ed.) (2021). *The Health Situation in Norwegian Aquaculture 2020*. Published by the Norwegian Veterinary Institute 2021. URL: https://www.vetinst.no/rapporter-og-publikasjoner/rapporter/2021/fish-health-report-2020.

Sommerset, Ingunn, Ceilie S Walde, Britt Bang Jensen, Jannicke Wiik-Nielsen, Geir Bornø, Victor Henrique Silda de Oliviera, Asle Haukaas, and Edgar Brun (2022). *The Health Situation in Norwegian Aquaculture 2021*. Published by the Norwegian Veterinary Institute 2022. URL: https://www.vetinst.no/rapporter-og-publikasjoner/rapporter/2022/fiskehelserapporten-2021.

Stien, Lars H., Jonatan Nilsson, Ernst M. Hevrøy, Frode Oppedal, Tore S. Kristiansen, Andreas M. Lien, and Ole Folkedal (2012). “Skirt around a salmon sea cage to reduce infestation of salmon lice resulted in low oxygen levels”. In: *Aquacultural Engineering* 51, pp. 21–25. ISSN: 0144-8609. DOI: https://doi.org/10.1016/j.aquaeng.2012.06.002. URL: https://www.sciencedirect.com/science/article/pii/S0144860912000647.

Svendsen, E. et al. (2021). “Heart rate and swimming activity as stress indicators for Atlantic salmon (Salmo salar)”. In: *Aquaculture* 531, p. 735804. ISSN: 0044-8486. DOI: https://doi.org/10.1016/j.aquaculture.2020.735804. URL: https://www.sciencedirect.com/science/article/pii/S0044848620302714.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2014). “Going Deeper with Convolutions”. In: *CoRR* abs/1409.4842. arXiv: 1409.4842. URL: http://arxiv.org/abs/1409.4842.

Szeliski, Richard (2011). *Computer vision algorithms and applications*. URL: http://dx.doi.org/10.1007/978-1-84882-935-0.

Titterton, David and John Weson (2004). *Strapdown Inertial Navigation Technology*. Radar, Sonar, Navigation. Institution of Engineering and Technology. URL: https://digital-library.theiet.org/content/books/ra/pbra017e.

Waite, A. D. (2001). *Sonar for Practising Engineers*. 3rd ed. ISBN: 0471497509; 9780471497509; 9780470867679; 0470867671.

Wang, Chien-Yao, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh (2019). “CSPNet: A New Backbone that can Enhance Learning Capability of CNN". In: *CoRR* abs/1911.11929. arXiv: 1911.11929. URL: http://arxiv.org/abs/1911.11929.

Wojke, Nicolai, Alex Bewley, and Dietrich Paulus (Sept. 2017). "Simple online and realtime tracking with a deep association metric". In: pp. 3645–3649. DOI: 10.1109/ICIP.2017.8296962.

Xing, Wei, Min Yin, Qing Lv, Yang Hu, Changpeng Liu, and Jiujun Zhang (2014). "1 - Oxygen Solubility, Diffusion Coefficient, and Solution Viscosity". In: *Rotating Electrode Methods and Oxygen Reduction Electrocatalysts*. Ed. by Wei Xing, Geping Yin, and Jiujun Zhang. Amsterdam: Elsevier, pp. 1–31. ISBN: 978-0-444-63278-4. DOI: https://doi.org/10.1016/B978-0-444-63278-4.00001-X. URL: https://www.sciencedirect.com/science/article/pii/B978044463278400001X.

Zaccone, G., M.R. Karim, and A. Menshawy (2017). *Deep Learning with TensorFlow*. Packt Publishing. ISBN: 9781786469786. URL: https://books.google.no/books?id=F9CMtAEACAAJ.

Zhu, Xiangxin, Carl Vondrick, Charless C. Fowlkes, and Deva Ramanan (2015). "Do We Need More Training Data?" In: *CoRR* abs/1503.01508. arXiv: 1503.01508. URL: http://arxiv.org/abs/1503.01508.

Appendices

Appendix A Block Diagram of FISC System

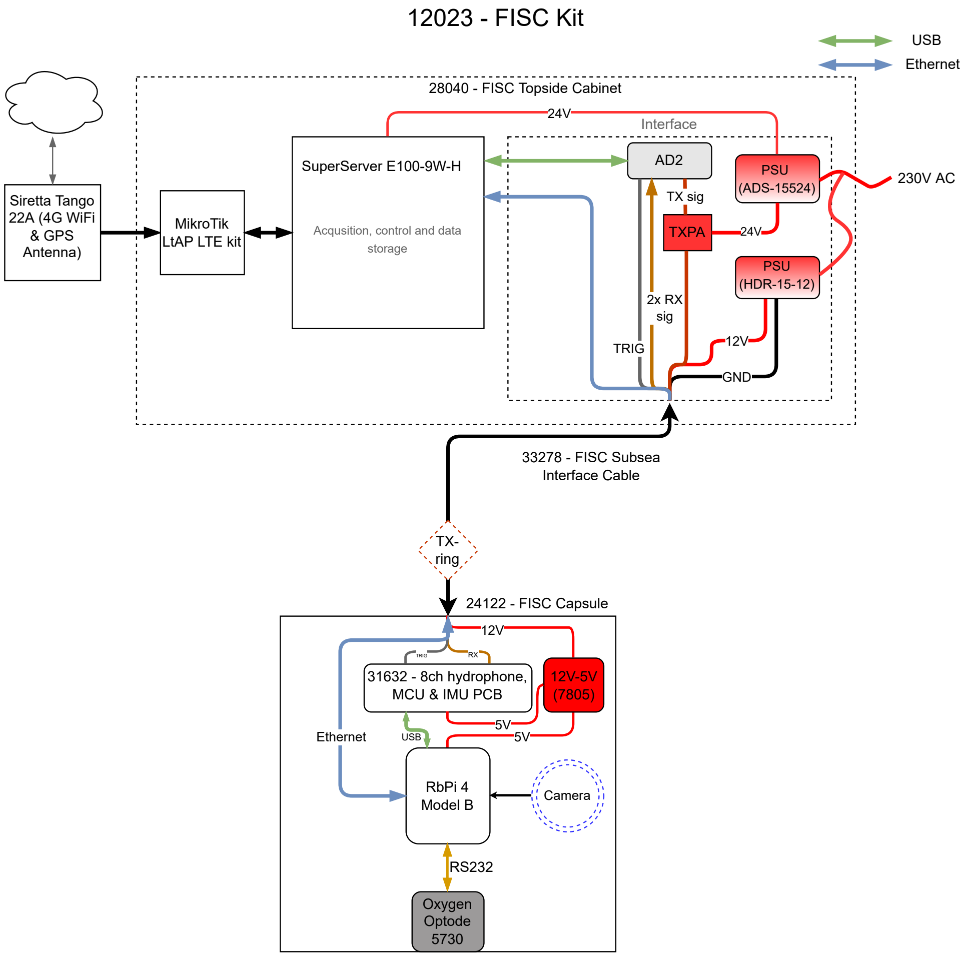


Figure A.1: Simplified block diagram of the FISC system.

Appendix B FISC Receiver Voltage Sensitivity

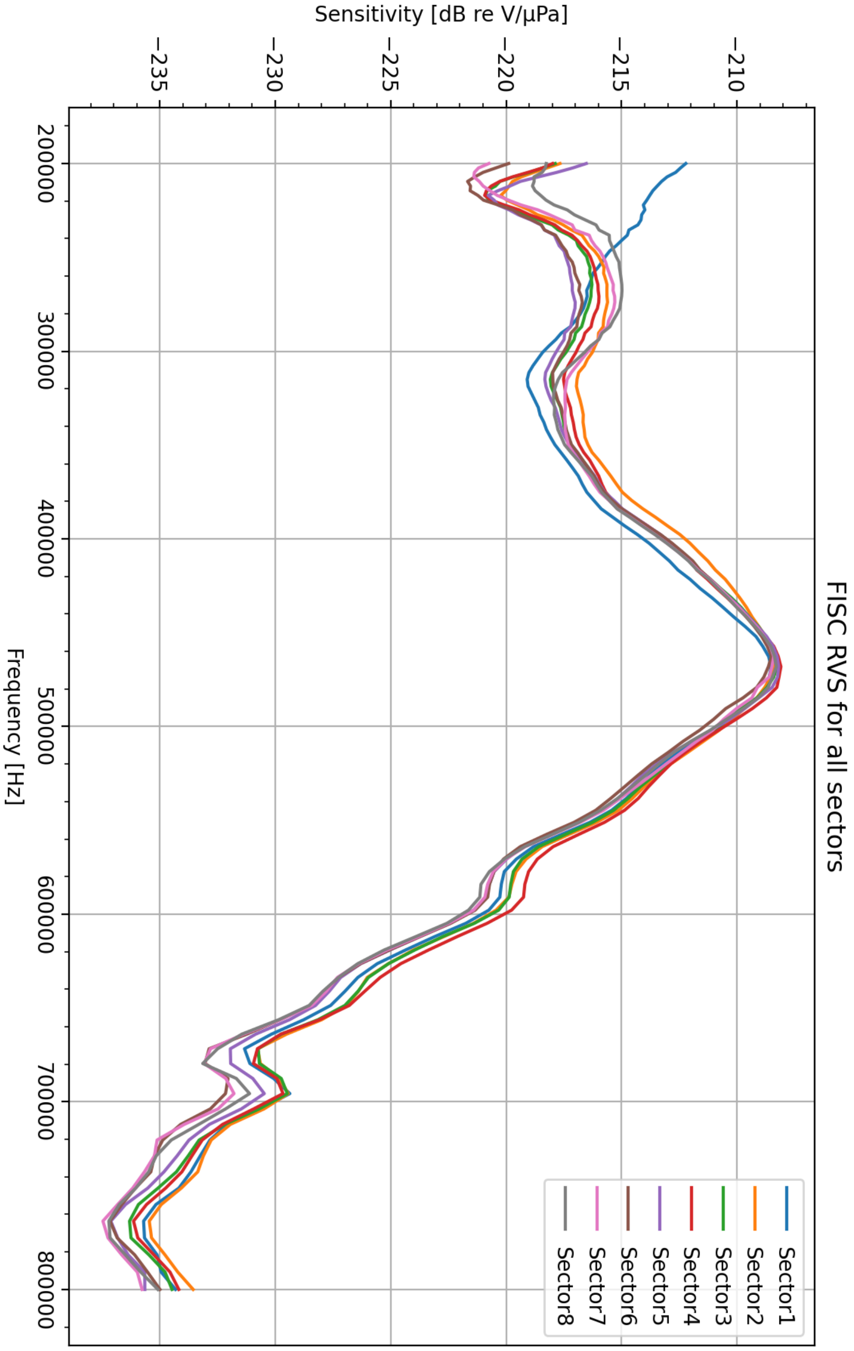


Figure B.1: FISC receiver voltage sensitivity for all eight sectors.

Appendix C FISC Receiver Directivity at 468 kHz

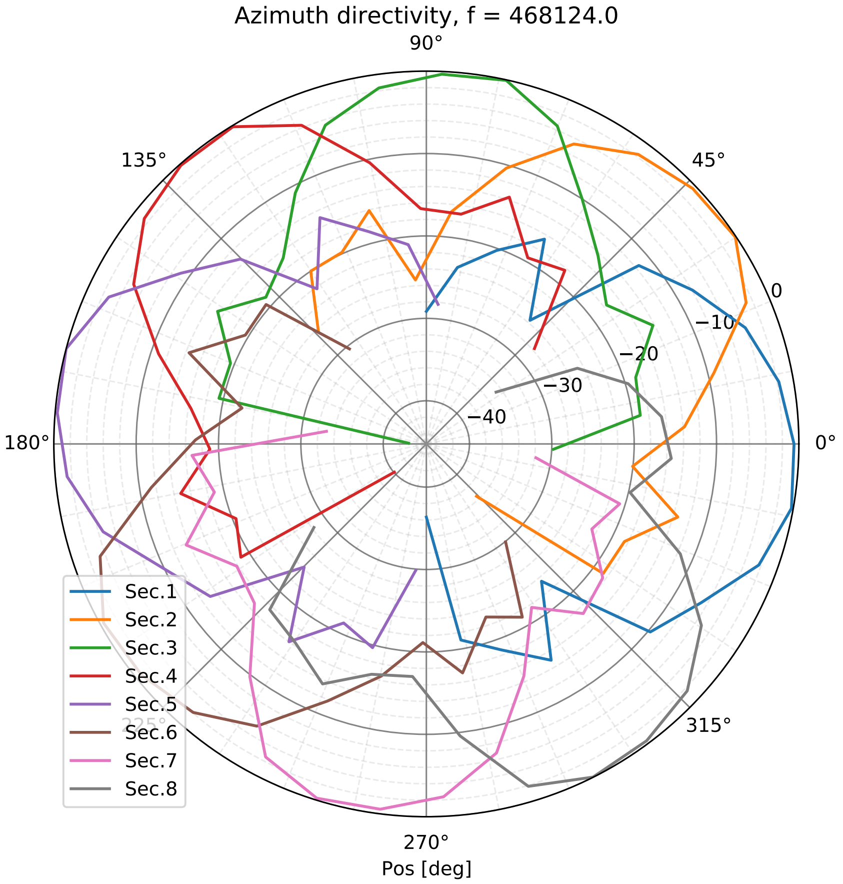


Figure C.1: FISC receiver directivity at 468 kHz.

Appendix D FISC TX Transmitting Voltage Response

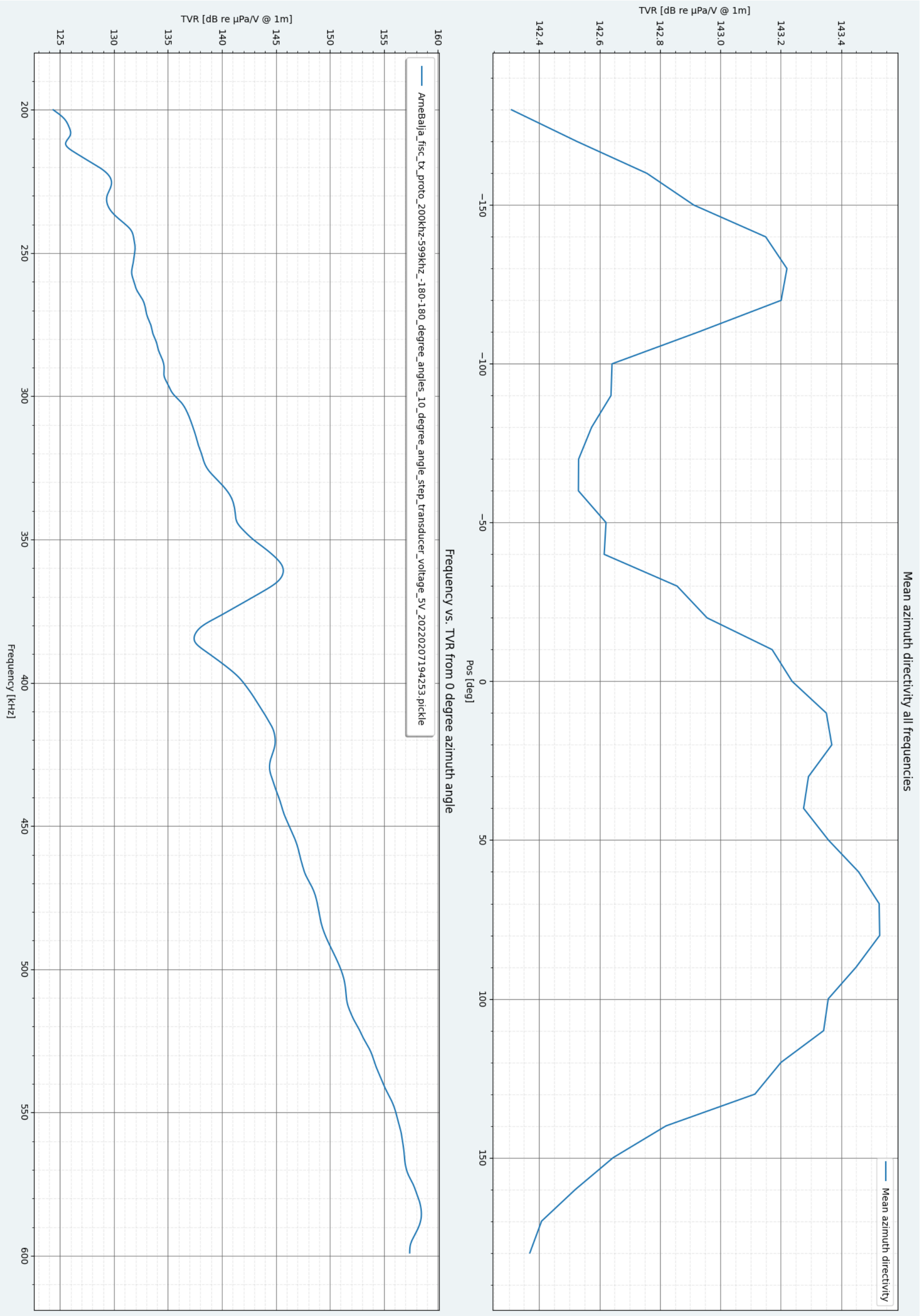


Figure D.1: FISC TX transmitting voltage response.

Appendix E FISC TX Directivity

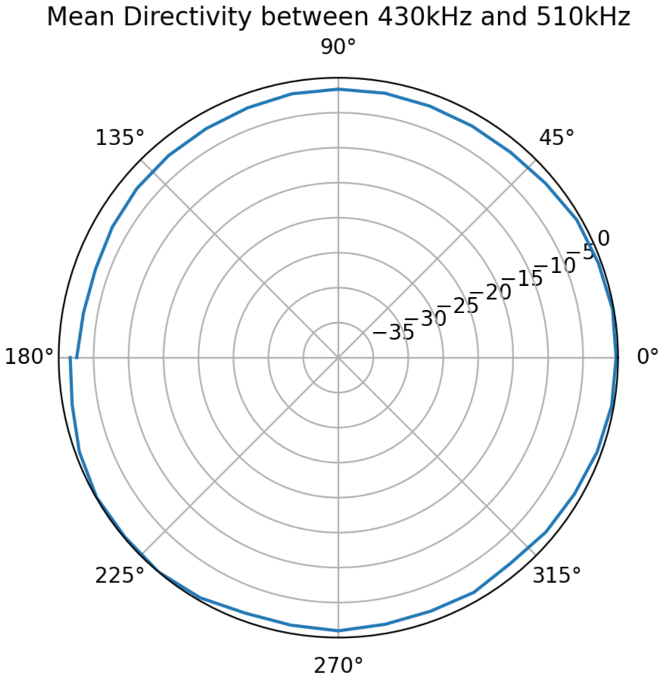


Figure E.1: FISC transmitter directivity.

Appendix F FISC Data-acquisition Software Flowchart

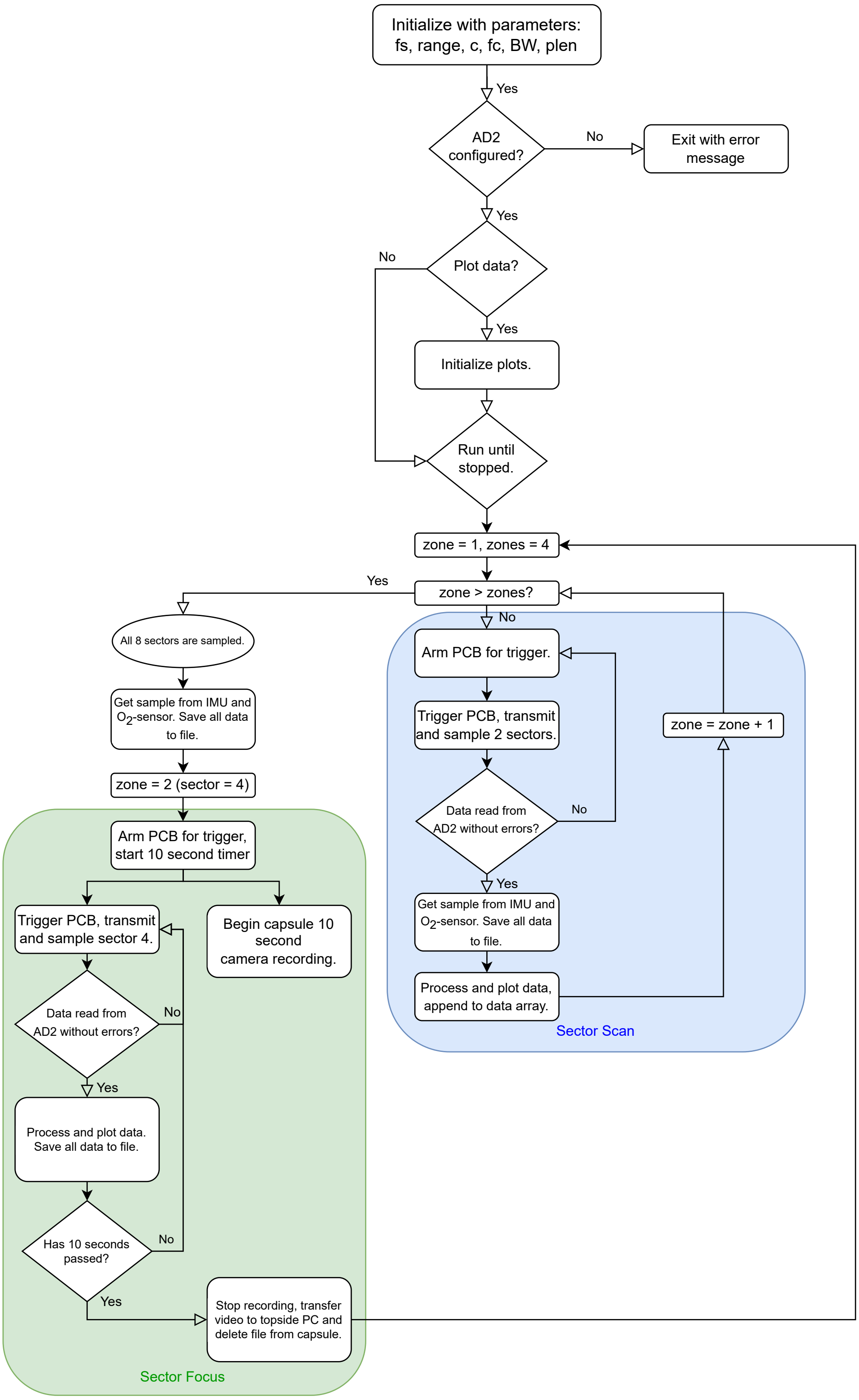
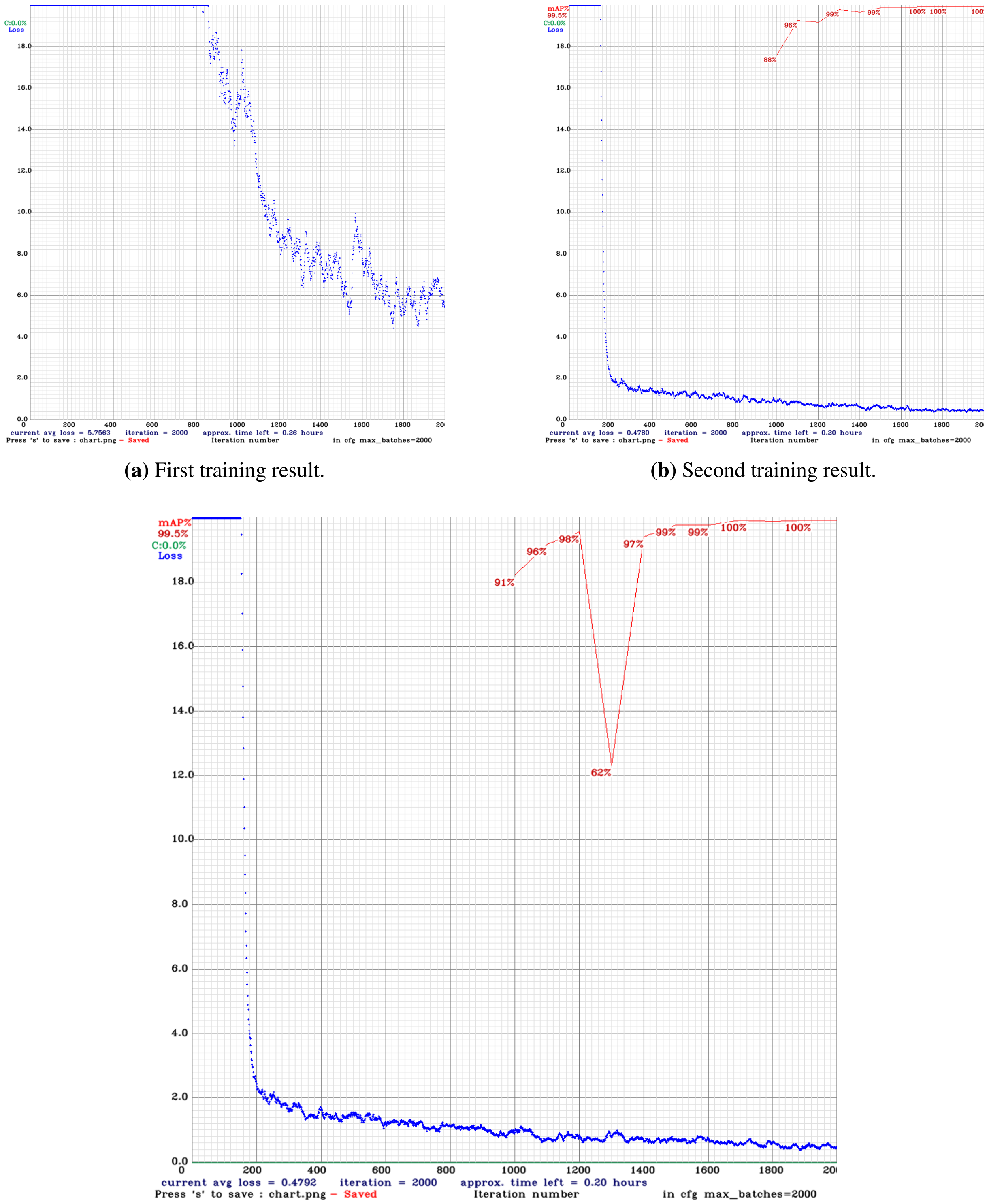

Initialize with parameters: fs, range, c, fc, BW, plen
Yes
AD2 configured?
No
Exit with error message
Yes
Plot data?
No
Yes
Initialize plots.
Run until stopped.
zone = 1, zones = 4
zone > zones?
Yes
No
All 8 sectors are sampled.
Get sample from IMU and O_2-sensor. Save all data to file.
zone = 2 (sector = 4)
Arm PCB for trigger, start 10 second timer
Trigger PCB, transmit and sample sector 4.
Begin capsule 10 second camera recording.
Data read from AD2 without errors?
No
Yes
Process and plot data. Save all data to file.
Has 10 seconds passed?
No
Yes
Stop recording, transfer video to topside PC and delete file from capsule.
Sector Focus
Arm PCB for trigger.
Trigger PCB, transmit and sample 2 sectors.
Data read from AD2 without errors?
No
Yes
Get sample from IMU and O_2-sensor. Save all data to file.
Process and plot data, append to data array.
zone = zone + 1
Sector Scan

Figure F.1: Simplified FISC data-acquisition software flow chart, adapted from Ericsson (2021).

Appendix G All YOLOv4 Training Results



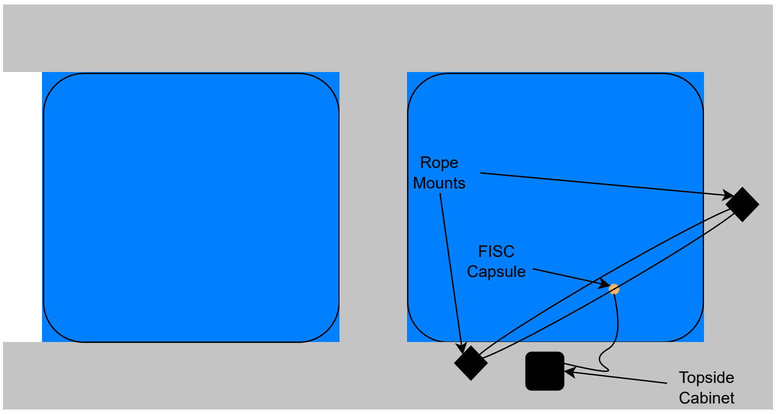
(**a**) First training result.

(**b**) Second training result.

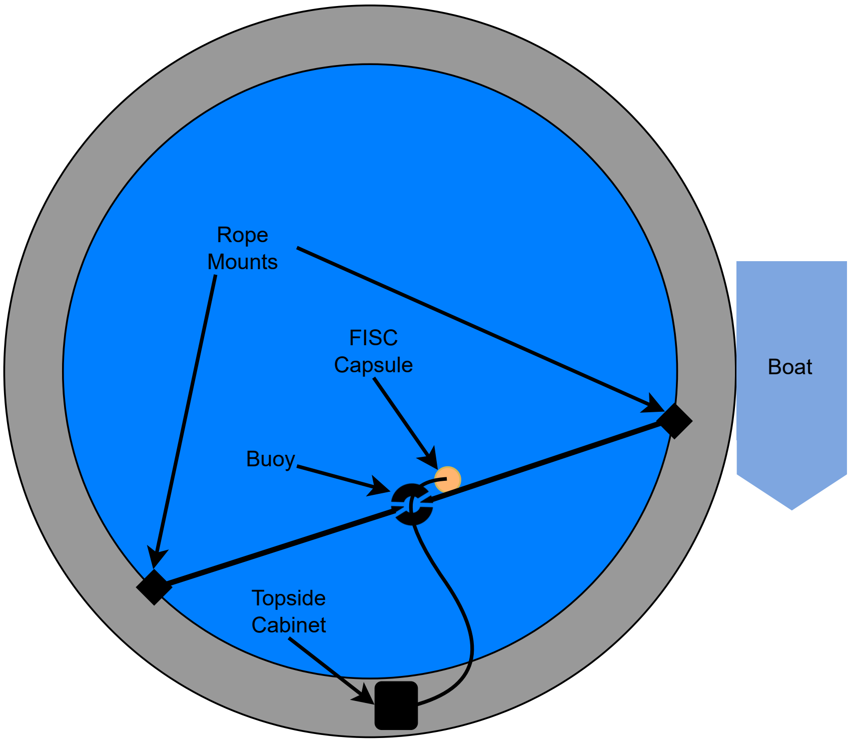
(**c**) Final training result.

Figure G.1: All YOLOv4 custom training results.

Appendix H Field Test Installation Illustrations



(a) Waiting-pen data acquisition installation (Sinkaberg Hansen AS).



(b) Rataren II data acquisition installation (SINTEF ACE).

Figure H.1: Illustrations of field test installations.

Appendix I Image of FISC Capsule from Second Field Test

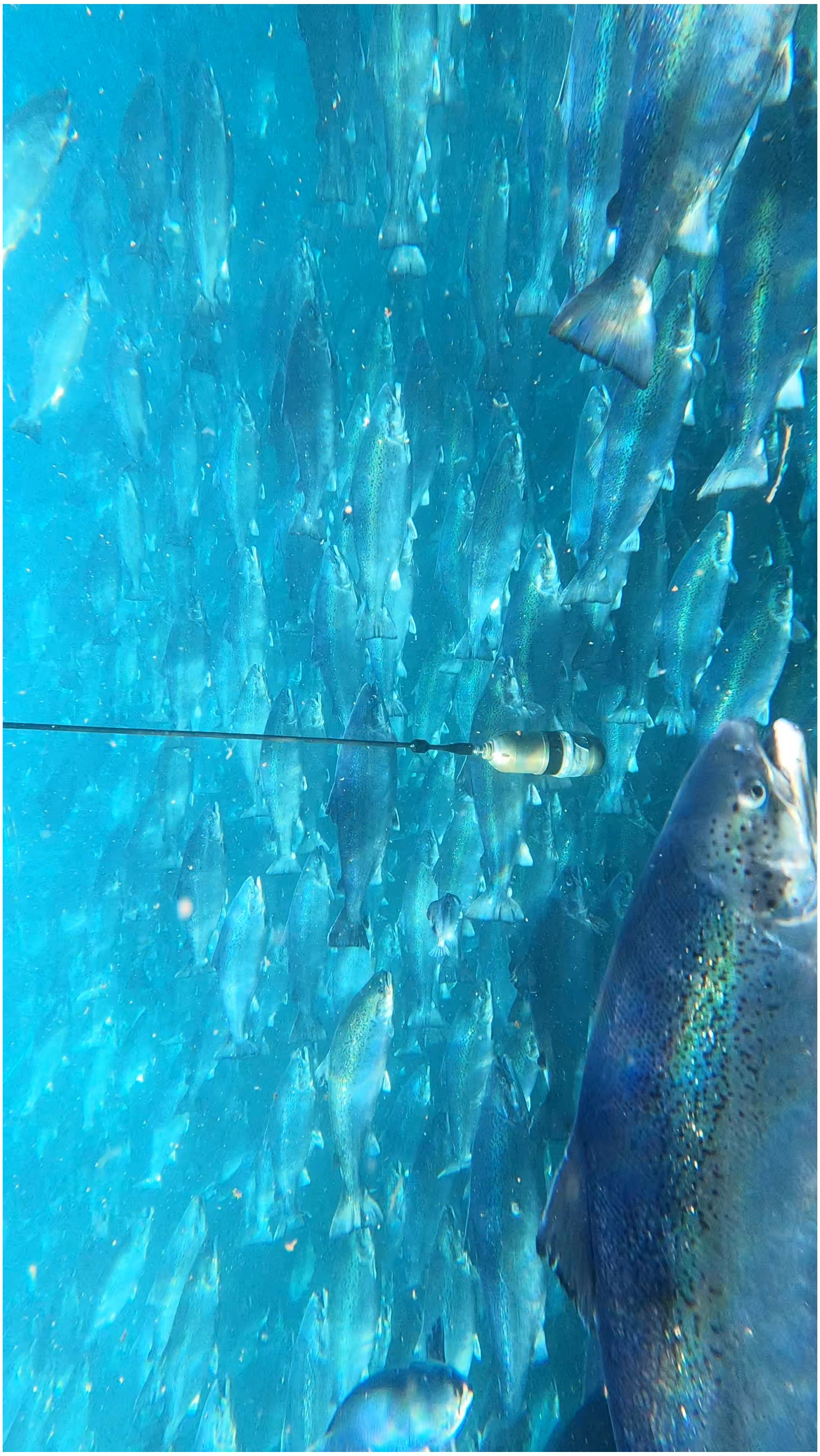
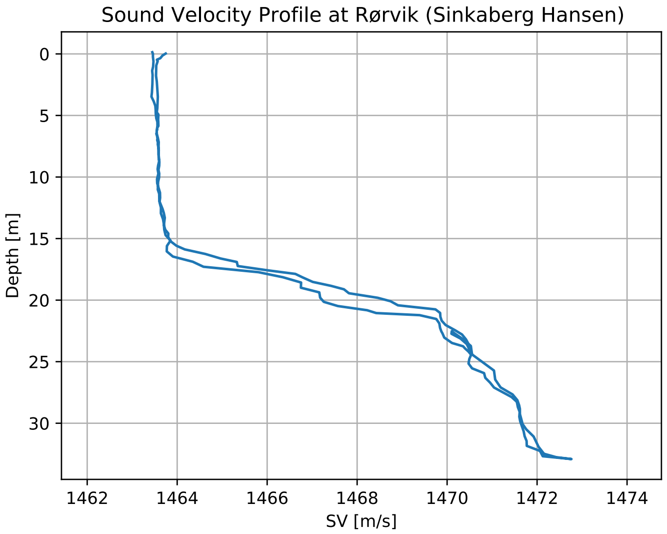
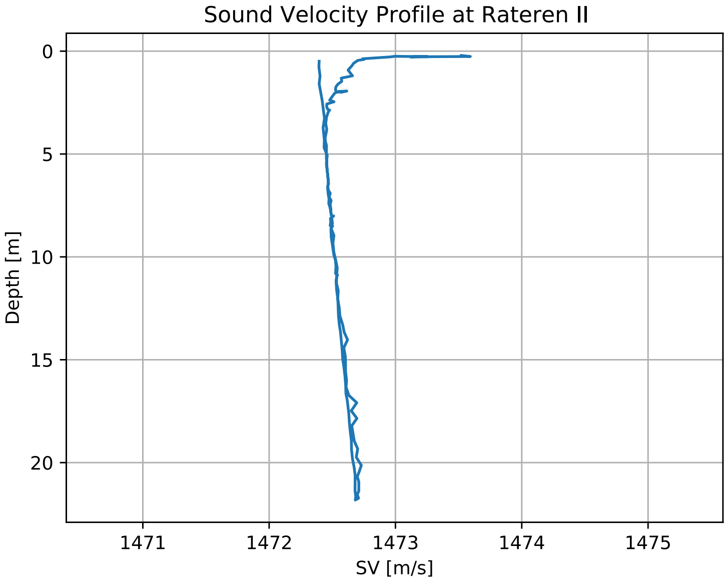


Figure I.1: GoPro image of FISC capsule in pen.

Appendix J Sound Velocity Profiles During Field Tests



(a) Sound velocity profile during initial field test.



(b) Sound velocity profile during second field test.

Figure J.1: Sound velocity profiles from both field tests.

Martin Ericsson

Fish-farm Integrated Sensor Cluster (FISC) Master's Thesis

NTNU
Norwegian University of
Science and Technology