NTNU
Norwegian University of
Science and Technology

Stabilization of underwater camera

Thomas Norum Ur

Master of Science in Cybernetics and Robotics
Submission date: June 2017
Supervisor: Annette Stahl, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Summary

This thesis describes the development, implementation and testing of a full scale underwater camera system for surveillance purposes in aquaculture. The mechanical development was carried out using Solidworks, and the software implementation is based on ROS (Robotic Operating System), in which several open source libraries have been incorporated. A mathematical model of the camera system has been derived as well as a simulation tool in Matlab for simulation. Suspended from a single rope, the camera system is equipped with a water jet propulsion system that allows the yaw (heading) to be controlled by the use of a PID controller. A gimbal inspired mechanism enables control of the camera pitch (tilt). Experiments at a full scale fish farm facility yields promising results for the yaw-control, whereas the pitch control needs to be further developed.

Preface

The work presented in this thesis has been carried out during the spring of 2017 in colaboration with Sealab Ocean Group and SINTEF Ocean. I would like to thank my superviser Annette Stahl for her help and support during the last year, as well as my co-supervisors Per Rundtop and Christian Schellewald at SINTEF Ocean. Further, I would like to thank Oscar Marcovic and Milan Marcovic for making the whole project possible and assisting me during the work. This project spans over several fields of engineering and has been both challenging and very educational.
Some parts of this thesis contains sections that are directly copied from my Specialization Project report ”TTK4550” on the same topic.

Table of Contents

List of Tables

List of Figures

Abbreviations

IMU	=	Inertial Measurement Unit
ESC	=	Electronic Speed Controller
PID	=	Proportional Integrator Derivative
BEC	=	Battery Eliminator Circuit
GPIO	=	General Purpose Input Output
ROS	=	Robotic Operating System
DOS	=	Degrees Of Freedom
CAD	=	Computer Aided Design

Chapter 1

Introduction

1.1 Problem formulation

A full scale underwater camera system capable of controlling yaw and pitch is to be developed, implemented and tested. The system should be able to track a referance trajectory in both yaw and pitch, and acquire stable images, free of shaking while doing so. Experimental tests sholud be carried out under controlled environment lab facilities, and at a full scale fish farm in order to evaluate the preformance of the system. In addition, a mathematical model equation describing the fundamental dynamics of the system is to be derived for simulation purposes.

1.2 Motivation

Aquaculture involves cultivating populations under controlled conditions, with fish farming being its most common form. Production of seafood has increased steadily over the last five decades and as of 2009, more than 50% of seafood was produced by aquaculture[1]. The aquaculture industry is facing major challenges that must be apprehended in order to ensure the quality and well being of the fish, while keeping the production at a responsible and sustainable level for present and future generations.

Addressing these issues, requires comprehensive video surveillance in the fish cages so that the proper analysis of the fish welfare can be carried out. Great advances have been made in recent years in the field of machine vision, but in order to utilize these methods in an efficiant manner, the camera systems aquiring the images must be enhanced. Fish farms are often located in areas along the coast with rough weather conditions, making it hard to deploy cameras that are capable of collecting images free of unwanted movement and shaking.

1.3 Report outline

Chapter 1 An introduction to the aquaculture industry and a motivation for solving the task at hand.

Chapter 2 Derivation of the equation of motion, and a description of the implemented controller.

Chapter 3 A description of the development and implementation of the camera system

Chapter 4 Experimental procedures described.

Chapter 5 Presentation and discussion of the results.

Chapter 6 The thesis is concluded.

Chapter 7 Further work.

Chapter 2

Theory

In this chapter, a mathematical model of the purposed underwater camera system is derived, based on methods presented in the **Handbook of Marine Craft Hydrodynamics and Motion Control** (Fossen, 2011)[2]. This book includes state of the art guidance, navigation and control systems for underwater vehicles that are highly applicable for the problem at hand.

The camera system, being an underwater structure, holds a lot of similarities with an underwater vehicle, such as a submarine. Therefore, the chosen approach was to use model equations for underwater vehicles as a foundation, and apply modifications so that the the resulting model exhibits the correct behavior.

2.1 Assumptions

Several assumptions and simplifications in the model are made. Firsly, the camera system is assumed to be one rigid body shaped as a cylindar representing the aquapod. Thereby, the camera house and it's ability to rotate relative to the aquapod, is not directly represented in this model.

Secondly, the rope from which the camera system is suspended, is modeled as a spring according to Hookes law, further described in section 2.3.4.

2.2 Kinematics

In order to derive the model equations, we must first establish a gemoetrical framework. That is, a set of coordinate systems and the gemoetrical relations between them.

A very commonly used base frame is the **NED** (North East Down) coordinate system, where the x-axis is pointed towards the north pole of the earth, the y-axis pointed east and the z-axis pointed towards the center of the earth. Seeing as the purpose of the model equation is not to navigate around the whole world, but to navigate inside a relatively small space such as a fish cage, the NED-frame is assumed to be earth fixed.

This also implies that the world is assumed to be flat, hence the name **flat earth navigation**.The second frame is the **BODY** frame which is fixed to the aquapod and has it's origin placed at the center of gravity of the aquapod. Figure 2.1 illustrates the frame assignment.

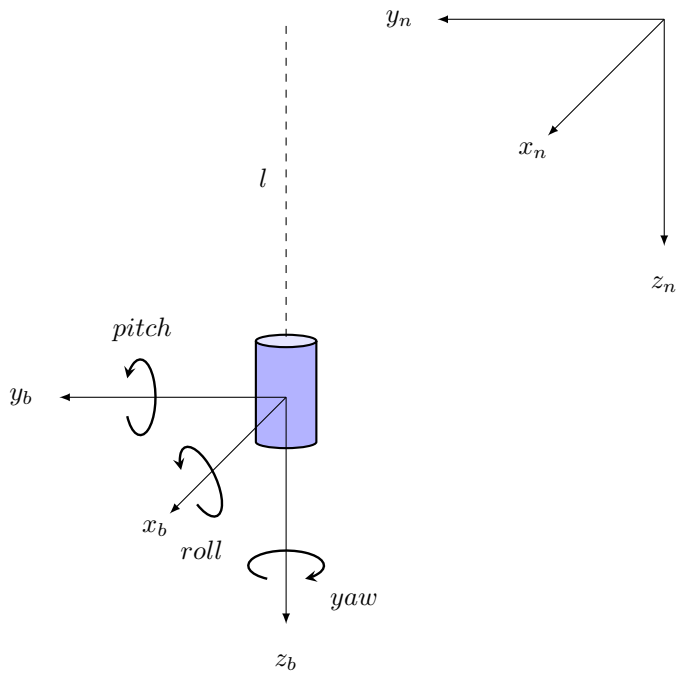


Figure 2.1: Frame assignment. NED frame $n = (x_n, y_n, z_n)$ and BODY frame $b = (x_b, y_b, z_b)$. Euler angles and their positive definition. l is the length of the rope.

Table 2.1 gives an overview of the notation used in the model.

DOF		Forces and momentums	Linear and angular velocities	Positions and Euler angles
1	motions in the x direction (surge)	X	u	x
2	motions in the y direction (sway)	Y	v	y
3	motions in the z direction (heave)	Z	w	z
4	rotation about the x axis (roll)	K	p	ϕ
5	rotation about the y axis (pitch)	M	q	θ
6	rotation about the z axis (yaw)	N	r	ψ

Table 2.1: The notation of SNAME[3]

2.3 Equations of motion

In order to express the equations of motion, the following notation in accordance with [2] is defined.

$$
\begin{array}{lll}
f_b^b & = [X, Y, Z]^T & \text{- forces through } o_b \text{ expressed in } b \\
m_b^b & = [K, M, N]^T & \text{- momentum about } o_b \text{ expressed in } b \\
v_{b/n}^b & = [u, v, w]^T & \text{- linear velocity of } o_b \text{ relative } o_n \text{ expressed in } b \\
\omega_{b/n}^b & = [p, q, r]^T & \text{- angular velocity of } b \text{ relative to } n \text{ expressed in } b \\
r_g^b & = [x_g, y_g, z_g]^T & \text{- vector from } o_b \text{ to CG expressed in } b
\end{array}
$$

The equations of motion can now be expressed in a vectorial setting as follows.

$$M\dot{\nu} + D(\nu) + g(\eta) + T(\eta) = Bu + \tau_{env} \tag{2.1}$$

where $\eta = [x, y, z, \phi, \theta, \psi]^T$ is the generalized position vector expressed in n, and $\nu = [u, v, w, p, q, r]^T$ is the generalized velocity vector expressed in b. Each term in equation 2.1 are further derived in the following sections.

2.3.1 Inertia matrix

The first term in the equations of motion accounts for the Newtonian rigid body kinetics, where M is the rigid body inertia matrix, and $\dot{\nu}$ is the time derivative of the generalized velocity vector. Given the assumption that the center of gravity is located at the origin of the BODY frame, o_b, the inertia matrix is defined as follows.

$$M = \begin{bmatrix} mI_{3\times 3} & 0 \\ 0 & I_b \end{bmatrix} \tag{2.2}$$

where m is the mass of the aquapod, $I_{3\times 3}$ is the Identity matrix, and I_b is the inertia tensor defined as

$$I_b = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yx} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{2.3}$$

Assuming that the principal axes of the cylinder is aligned with the axes of the BODY frame, the non diagonal entries are all zero, leaving only the moments of inertia about x_b, y_b and x_b. The Inertia matrix can be written out as

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{xx} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{yy} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{zz} \end{bmatrix} \tag{2.4}$$

2.3.2 Restoring forces

The term $g(\eta)$ accounts for the restoring forces exterted by gravitation and by bouancy due to water dissplacement, denoted f_g^n and f_b^n respectively in n.

$$f_g^n = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad and \quad f_b^n = - \begin{bmatrix} 0 \\ 0 \\ \rho g V \end{bmatrix} \tag{2.5}$$

where m is the mass of the aquapod, g is the gravitational constant, ρ is the desity of the water, and V is the volume of the aquapod. Under the assumption that center of boyancy is located at the origin of the BODY frame, $g(\eta)$ can be written out in b as follows.

$$g(\eta) = \begin{bmatrix} f_g^n + f_b^n \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.6}$$

2.3.3 Drag forces

Drag force is the friction that exists between the surface of the aquapod and the surrounding fluid (water), and acts in the opposite direction of the velocity of the aquapod relative to the fluid. For an underwater vehicle in 6 DOF, the drag forces are typically nonlinear and highly coupled. However, as suggested in [2], this can be roughly approximated as follows.

$$D(\nu) = \begin{bmatrix} -sign(u)\frac{1}{2}\rho A C_u u^2 \\ -sign(v)\frac{1}{2}\rho A C_v v^2 \\ -sign(w)\frac{1}{2}\rho A C_w w^2 \\ -sign(p)\frac{1}{2}\rho A C_p p^2 \\ -sign(q)\frac{1}{2}\rho A C_q q^2 \\ -sign(r)\frac{1}{2}\rho A C_r r^2 \end{bmatrix} \tag{2.7}$$

where the function $sign(i)$ returns the sign of i, ρ is the density of the water, A_i is the projected area of the cylinder on a plane perpendicular to the direction of the motion. C_i denotes a dimensionless drag coefficient related to the shape of the cylinder.

2.3.4 Suspension

In a permanent scenario, the camera system will most likely be hanging from a rope, which in turn will be attached to a second rope suspended horizontally across the fish cage, forming a T-shaped suspension arrangement. This can be modeled fairly accurate as a spring that can only pull the camera system up, but never push it down (as an actual spring would have done). The length of the rope is defined as l, and the force f_r produced by the rope is defined according to Hookes Law as

$$f_r = \begin{cases} k(|p_b^n| - l) & \text{if } |p_b^n| > l \\ 0 & \text{if } |p_b^n| <= l \end{cases} \tag{2.8}$$

where k is a characteristic constant describing the stiffness of the spring, and $|p_b^n|$ is the length of the vector describing the position of the origin of b relative n. The direction in which f_r is applied is determined by the position vector p_b^n. The suspension matrix $T(\eta)$ is defined as forllows.

$$T(\eta) = \begin{bmatrix} f_r \frac{p_b^n}{|p_b^n|} \\ r_r^n \times (f_r \frac{p_b^n}{|p_b^n|}) \end{bmatrix} \tag{2.9}$$

where r_r^n is a vector from the origin of the BODY frame to where the rope is attached to the aquapod.

2.3.5 Simulation

A state space representation of the model as stated above has been implemented in a MATLAB environment for simulation purposes. The simulation script can be found in the digital appendix. The simulation script can be found in the digital appendix.

2.4 PID Control

There are two actuated degrees of freedom in the full scale implementation in need of a controller. This section describes the inner workings of the controller that is implemented in software in the prototype, which is a PID-controller (Proporti nal Integral Derivative). For convenience, the following description is concerned with the yaw-controller, but it works exactly the same for the pitch controller.

A PID-Controller is a feedback loop mechanism used to achieve a desired behavior of a given process, in this case; the angular positions (yaw and pitch) of the camera system. To further describe the workings of the feedbacklaw, we must first establish some variables.

$\psi(t)$	*angular yaw position (heading)*
$\psi_d(t)$	*desired angular yaw position*
$e(t) = \psi(t) - \psi_d(t)$	*error*
$u(t)$	*control variable (applied torque)*

The algorithm starts with a measurement of the current yaw angle followed by a comparison with the desired yaw angle to compute the error. Based on this error, the control variable is computed as a weighted sum as follows.

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{d}{dt} e(t) \tag{2.10}$$

The control variable is composed of three terms, each with its own gain coefficient. The first one is called the proportional term and accounts for the present error. As the name suggests, this term is proportional to the error. The value of K_p is related to the responsiveness of the controller. A larger K_p will make the controller faster, but it comes at a cost. An oversized K_p will result in what's called an overshoot, and eventually oscillations. The second term is called the integral term, and accounts for the past errors. The longer an error is present, the larger this term will grow. The main purpose of this term is to eliminate steady-state errors. The third term is called the derivative term. It's purpose is to predict future behavior of the system based on the errors current rate of change. If the error is rapidly approaching zero, this term will "turn on the brakes" before the error reaches zero, preventing the system from overshooting.

Every dynamical system will behave differently. In order for this algorithm to work optimally, the coefficients K_p, K_i and K_d need to be tuned to fit the application in question.

Chapter 3

Implementation

During the course of this project, a full scale prototype of the suggested underwater camera system has been designed and implemented. This chapter is divided into three sections, describing various of the process. The first section is concerned with the mechanical implementation and contains both computer generated CAD-models and pictures of the prototype during assembly. The second section gives a brief description of the hardware components used and how they work together. The software is implemented using Robotic Operating System (ROS), and is the topic of the last section.

3.1 Mechanics

There are a great number of considerations to be taken into account when designing a submersible camera system. Spacing inside watertight enclosures are limited, so every component has been carefully chosen to make it all fit together. This chapter describes the process of designing and building the system.

The concept of using two watertight enclosures to house the camera and other electronic devices is adopted from a similar prototype developed at Sealab AS. Having repeatedly proven to be a robust and reliable solution, the same approach was taken for the system at hand. The camera house was replicated at the beginning of this project and served as the base upon wich the rest of the system was designed.

The entire system is designed in Solidworks, a Computer Aided Design(CAD) software. Solidworks is one of the most widely used CAD softwares amongst designers and engineers, this may one of be the reason why it is very common for technology producers to offer a CAD-model of their products on their web sites. This way one can easyly see wether or not the product in question is going to fit into the assembley, before bying it.

3.1.1 Camera House

A cylinder shaped aluminum piece is fitted with a glass dome, creating the exterioir of the house. The back lid is equipped with a mounting braket that holds the camera in place and has been modified for this project so that it also holds an IMU sensor, facilitating real time measurements of the cameras attitude.

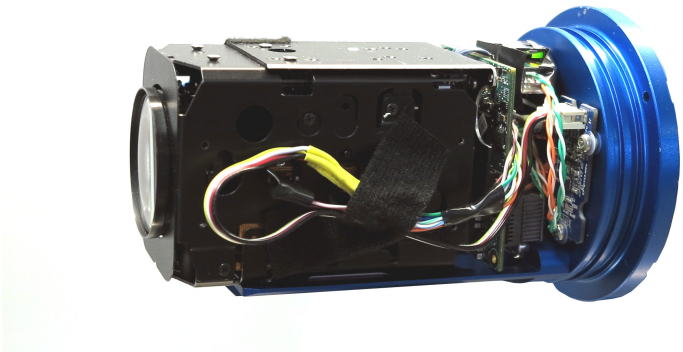


Figure 3.1: Camera and IMU mounted onto back lid of camera house

3.1.2 Aquapod

Holding most of the electronics, the Aquapod serves as control center. The aluminum body was originally used in a preexisting prototype and has been redesigned and modified to fit the needs of this project. Shaped like a cylinder with an inner diameter of about 8 cm, installing the hardware becomes rather difficult. This was solved by designing the top cap as a tower onto which the electronics can be mounted. Figure 3.2 illustrates the concept.

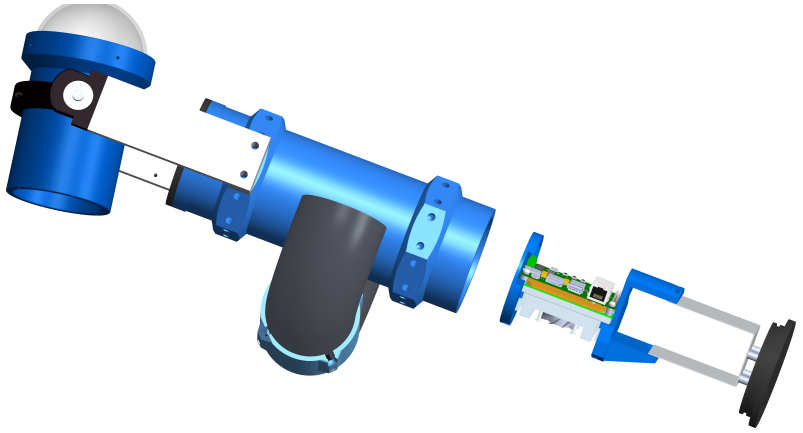


Figure 3.2: Aquapod assembley (CAD-model)

Four screws mounts the lower part of the tower onto the cap. Most of the components are bolted onto the aluminum scheleton. The largest device in the Aquapod is the Jetson TX1 which is located at the top of the tower. Two 3D-printed brackets have been tailored to fit the mounting holes in the TX1, while perfectly matching the inner diameter of the Aquapod and maintaining concentricity with the cap. This makes for a nice guidance in the final assembley and keeps the tower fixed inside the aquapod. Once all the electronics are in place and wired correctly, the tower can be slided into the cylinder.

Figure 3.3 depicts the final assembley of the top cap and tower with all components installed and wired. Five incoming cables run through penetrators in the cap, leaving the sixth free for a potential expansion. In the event of a modification or troubleshooting, the components are made highly acecssible simply by extracting the cap/tower from the aquapod. When the top cap and aquapod are mated, a satic radial seal ensures that the installation is watertight.

Machine drawings were made from the CAD model of the top cap and machined in Polyoxymethylene(POM) at the Department of Engeneering Cybernetics.

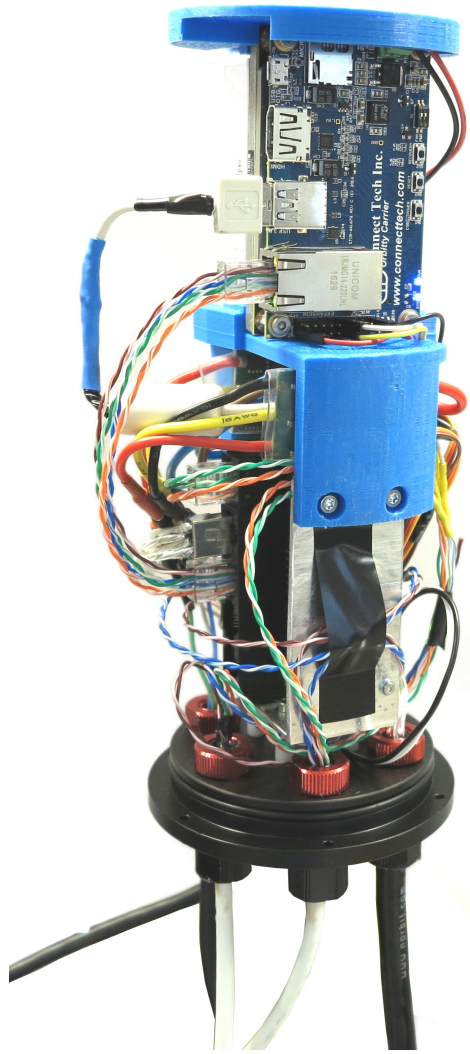


Figure 3.3: Aquapod tower assembley

3.1.3 Yaw Actuation

One of the key features of the camera system, is the ability to control it's heading. In order to achieve this, a torque must be applied about the z-axis of the aquapod. A thruster is placed inside a tube with a 90 degree bend on each end, effectively converting the linear force produced by the thruster into angular torque. The thruster itself is extracted from it's original frame and placed inside a 3D-printed bracket as shown in the exploded view of the assembley in figure 3.4. After the thruster is slided into the bracket and fastened with screws, a 90 degree bend is carefully positioned and fastened with two component epoxy at each end of the bracket. Finally the subassembley is mated with the aquapod using silicone adhesive and cable ties.

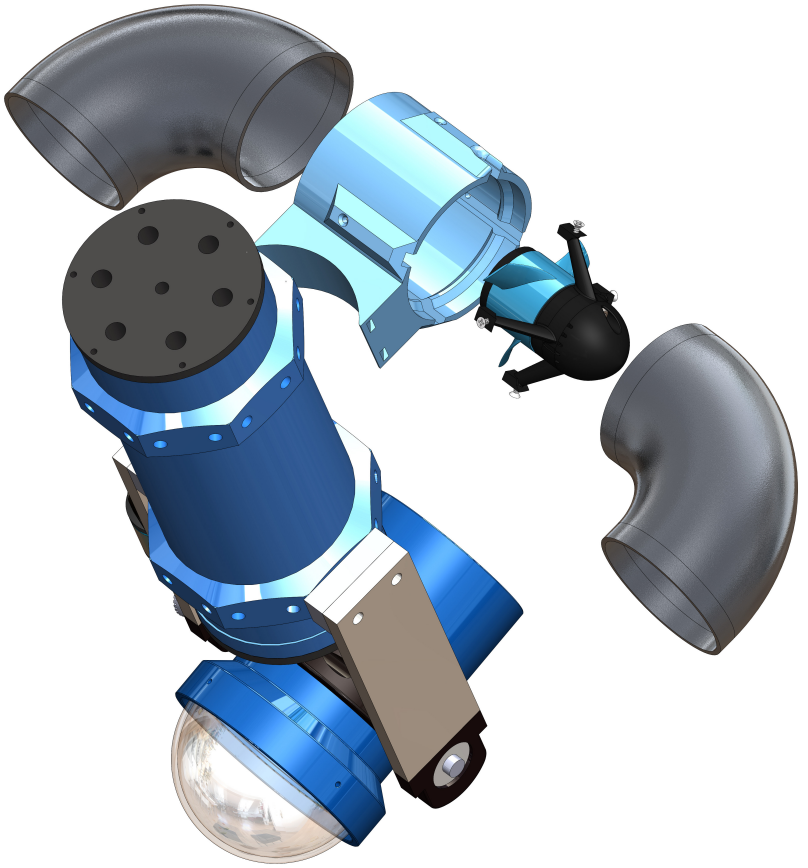


Figure 3.4: Water Jet Propulsion, exploded view

3.1.4 Pitch Actuation

Figure 3.5 illustrates the mechanism that allows the cameras pitch to be actuated. The camera house is enclosed by a 3D-printed clamp that's fitted with circular slot in each end where a short shaft is mounted. The shafts are perfectly aligned and runs straight through the center of the camera house cylinder. The shaft is resting inside a nylon bering with glass bearing balls, made espessially for underwater use. A brushless DC motor is mounted vertically onto the aluminum arm and is geared down with a 1:4 ratio molded POM bevel gear.

The center of gravity and center of bouyancy of the camera house is assumed to coincide with the axis of pitch rotation, effectively balancing the static pitch torque. This assumtion not likely to be true, but the clamp holding the camera house provides the possibility of experimentally finding the optimal position that satisfies this assumtion to the best extent.

Figure 3.5: Pitch actuator, exploded view

3.2 Hardware

In this section, a small description of each of the components in the camera system is given. The main architecture of the components and the information flow is illustrated in figure 3.6

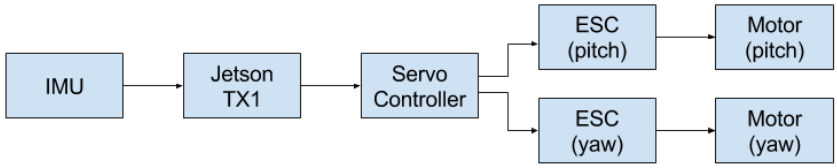


Figure 3.6: Information flow between main components.

3.2.1 Sony FCB-EV7520

The Sony FCB EV-7520 industry camera is capable of capturing crisp, clear Full HD (1080/60p) images [4]. Raw image material from the camera is processed on the Ionodes Atomas Mini camera interface, which is mounted on the back of the camera. The camera system is powered by a 12VDC-supply and the videofeed is transmited over ethernet. Utilizing POE (Power Over Ethernet), both needs are covered with one CAT-5 cable. Wiring diagram for this cable can be found in the attached digital appendix. (TO DO: MAKE DIAGRAM!!!!)

3.2.2 Inertial Measurement Unit

The IMU-sensor used for this project is a **Grove - 9DOF IMU**. It provides measurements of linear acceleration, angular rate, and magnetic field in three axes using a combination of accelerometers, gyroscopes and a magnetometers. IMU's are typically used in Inertial Navigation Systems for calculation of position, attitude, velocity etc. By using data fusion algorithms, raw data from accelerometers and gyroscopes can produce accurate estimations of the cameras orientation relative an earth fixed frame (pitch, roll and yaw).

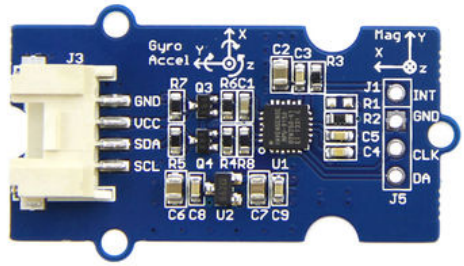


Figure 3.7: Grove -9DOF IMU.[5]

The magnetometer showed very little promising results during initial tests, which one could only expect concidering the electromagnetic environment inside the camera house submerged in water. Therefore, the magnetometer is not being used for the current configuration. The estimated yaw angle has no referance to true north, and the estimation will suffer from a small drift over time. This effect is concidered to be neglectable inside the scope of this project.

3.2.3 NVIDIA Jetson TX1

Jetson TX1 is a supercomputer on a module capable of delivering high preformance GPU and CPU computing, making it very well suited for robotics or any other small form factor environment. The TX1 is connected to a carrier board, facilitating the needed hardware interfaces such as power supply, ehternet, usb3.0 , GPIO etc. while keeping the size to a minimum. Figure 3.8 illustrates how the TX1 and the carrierboard are connected.

Figure 3.8: NVIDIA Jetson TX1 and Orbitty Carrier.[6]

3.2.4 T200 Thruster

The T200 Thruster, made by Blue Robotics, is designed specifically for marine robotics with a brushless DC-motor at it's core. The frame enclosing the thruster has been removed end replaced by a custom 3D-printed bracket as illustrated in figure 3.4.

Figure 3.9: Blue Robotics T200 thruster[7]

3.2.5 M200 Motor and ESC

The motor that runs the pitch actuation is the same motor that powers the T200 thruster.

Operating a brushless DC-motor requires a brushless Electronic Speed Controller. It's main task is to control which coil set to power, given the angular position of the rotor with respect to the stator. Seeing as the motor has no sensor to acurately measure this position, it is estimated based on the backemf induced by the coil set that is not powered at a given time.

3.2.6 Power Distribution

The electrical components in this system runs mainly at three different voltages, 5, 12 and 18 VDC. Power is distributed at 18 VDC from a topside power supply and transformed down to 12 and 5 VDC inside the aquapod using a Mini Power Hub with Battery Eliminator Circuit (BEC) as depicted in figure 3.10.

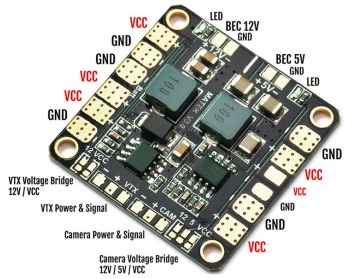


Figure 3.10: Matek Systems Mini Power Hub [8]

3.2.7 Cables

Each cable that runs from the outside of the structure and into either the aquapod or the camera house is fitted with epoxy filled penetrators that ensures protection against leakage. The umbilical is a 10-wire cable made specially for underwater use and consists of a CAT-6 ethernet cable and two copper wires for power supply.

3.3 Sofware

The Jetson TX1 is running a Ubuntu distribution made specifically for TX1. The software implementation is based on the Robotic Operating System (ROS), which is basically a framework for robot software development that makes for a highly modular development and provides a wide range of libraries. Message-passing between processes, even on different computers in a network is one of the great features of ROS and has proved to be very usefull in this project.

The yaw- and pitch-controllers are implemented in software on the NVIDIA Jetson TX1.

3.3.1 ROS architecture

Software in ROS is devided into **packages**, and within each package is a number of **nodes**. Each node is a separate executable that can be run completely independently from all other nodes. Message-passing takes place between nodes on a given **topic**,and the nodes can either publish or subscribe. Figure 3.11 illustrats the interaction between nodes implemented in the Jetson TX1.

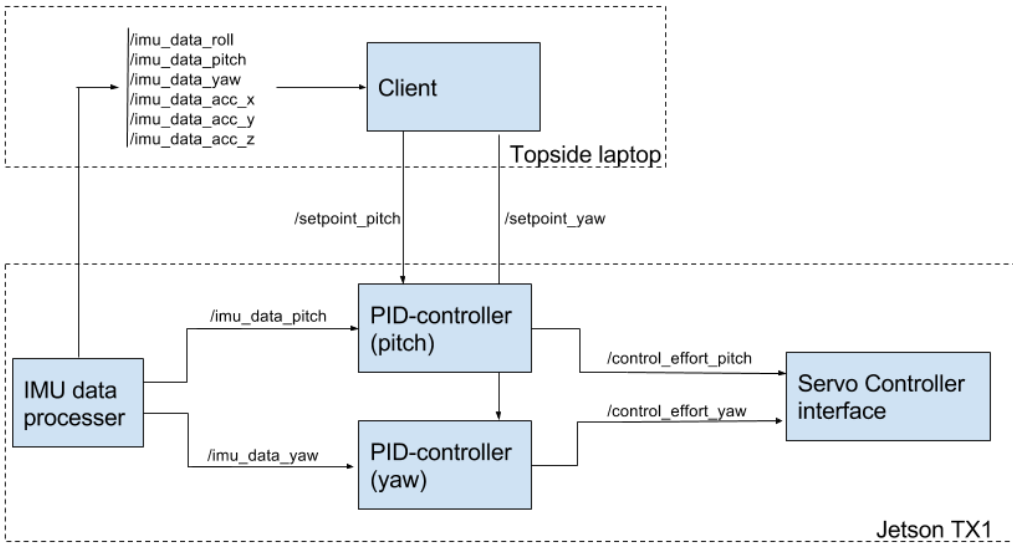


Figure 3.11: Ros architecture. Each block represents a node. Arrows represnts communication between nodes.

3.3.2 IMU interface

All interaction with the IMU takes place in the IMU interface node. It's main purpose is to feed the controllers with measurements of the yaw and pitch angle posistions. Seeing as the IMU it self can only provide raw data from accelerometers and gyroscopes, there is a need for a data fusion algorithm that can estimate the euler angles based on the raw

IMU data. To that end, a versatile 9DOF IMU Library for embedded Linux systems called RTIMULib[9] has been integrated.

Communication is established over the I2C protocol, and raw data from the accelerometer and gyroscope in three axes is pulled at a manufactor recommended rate (80 Hz) and passed through the data fusion filter that can be either a Kalman Filter or a simplified Kalman filter called RTQF. Furthermore, the estimated euler angles are published in the ROS environment on various topics as can be seen in figure 3.11.

3.3.3 Controller

A great number of open source packages are available in ROS, one of which is a general purpose implementation of the PID-controller alogorithm as described in section 2.4. The package has a lot of nice features that come in handy such as lowpass filters and dynamic configuration of the PID-parameters (K_p, K_i and K_d) in runtime to name a few. Two instances of the pid controller node are configured. One for yaw, and one for pitch. See ([10] for further details about the package and source code.

3.3.4 PWM interface

The general functionality of the Servo Controller Interface is fairly simple. It's job is to update the desired motor speeds commanded by the controllers. The control effort produced by the controller varying from -100 to 100 is remapped to the corresponding PWM-signal and transmitted to the Servo Controller unit over USB.

3.3.5 Client

As mentioned, one of the great features of ROS is the message-passing functionality. Any computer in the same network as the Jetson TX1 can easily subscribe and read out messages in real time on any given topic either in a linux terminal, or in a separate ros-node.

The client node illustrated to be running on a topside laptop in figure 3.11 is a node used during testing and it's only job is to publish a time varying setpoint for the yaw controller.

Chapter 4

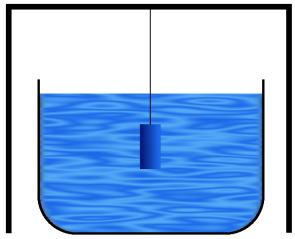
Experiments

In order to validate the preformance of the implementation, a number of experiments have been conducted.

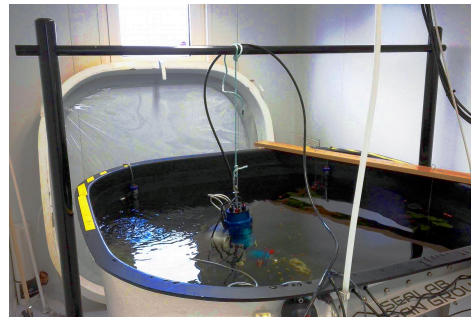
Initial testing took place under controlled environment at SEALAB's wet lab facilities located at Nyhavna, Trondheim.

4.1 Lab setup

Figure 4.1a illustrates the setup at the wet lab facilities at SEALAB. The camera system is suspended from a fixed steel bar and submerged in a tank filled with fresh water. The length of the rope is fixed throughout all tests at 1.0 meter. One of the advantages of using a setup such as this is the ability to reproduce the conditions under which the tests are conducted, espessially during the tuning process of the controllers. It's important to note that the rope from which the camera system is suspended, is not the only thing attatched to the camera system. As can be seen from the picture in figure 4.1b, the umbillical will start to twirl around the rope when rotated, thus introducing a small torque about the yaw axis.



(a) Illustration



(b) Lab setup at SEALAB wet lab

Figure 4.1

4.2 Field setup

The field test took place at a Marine Harvest fish farm facility located at Kåholmen, Hitra. Unlike most fish farms in Norway, this particular one has a large steel frame connecting all the cages, making them easily accessible.

The camera system is suspended from a ledge 3.0 m from the surface of the water, and submerged 2.0 m into the fish cage. Due to physical constraints concrening wiring and power outlets, the umbilical is not hanging parallel with the rope all the way up to the ledge, but is still loose enough that the torque introduced by twirling with the rope can be neglected within $\pm$ 180 degrees (yaw).

4.3 Step response

The objective of this test is to evaluate the preformance of the yaw PID-controller. For some time $t < 0$, the desired heading ψ_d is at a constant value, then at $t = 0$, it's instantly changed to another value. The resulting behavior reveals a number of properties regarding stability and the systems ability to reach a steady state when starting from another. The time it takes for the measured yaw angle ψ to reach the desired value of

90 degrees is defined as the setteling time and is a measure of the controllers speed. A speedy controller can be crucial if the system is to be used for tracking rapidly time varying trajectories.

Step response tests were performed both in the wet lab, and in full scale at Kåholmen.

4.4 Trajectory tracking

In this test the desired yaw angle ψ_d is a continious function of time in order to evaluate the controllers tracking abilities. The chosen function is based on the trigonometric function $arctan(t)$ with some adjustments. Spesifically it is defined as follows.

$$\psi_d(t) = \frac{2}{\pi} 45 arctan(1.1t) + 45 \tag{4.1}$$

which makes for a smooth transition from roughly 0 to 90 over the course of ten seconds. The function is plotted in figure 4.2.

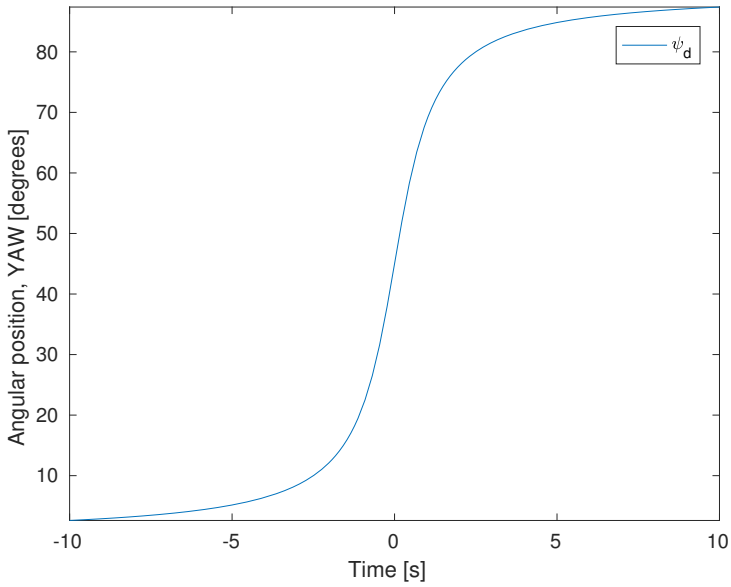


Figure 4.2: Desired yaw angle ψ_d in trajectory tracking test.

This test was performed both in the wet lab, and in full scale at Kåholmen.

4.5 Free fall test

The rope from which the camera system is suspended is held in the hand by a person, keeping the camera system positioned just below the surface of the water. The rope is

suddenly let go, leaving the camera system in a free fall deeper into the sea until the available rope length of three meters is up.

The data collected from the IMU during this test can be used to tune parameters in the equations of motion to increase the accuracy of the model.

This test was preformed at Kåholmen fish farm.

Chapter 5

Results and Discussion

5.1 Lab experiments

5.1.1 Step response

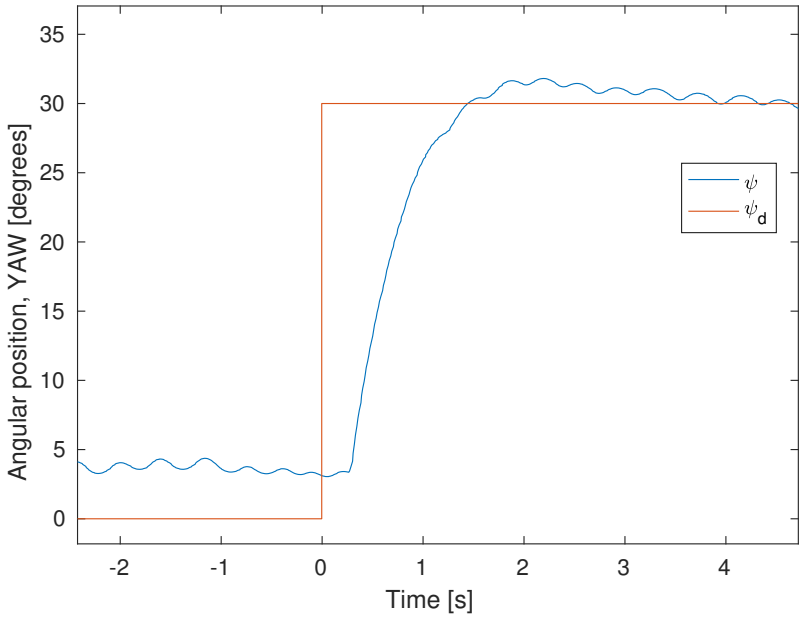


Figure 5.1: Step response, wet lab.

One of the first observations to be made is the presense of a small oscillation in the measurement of the yaw position. This is probably due to the fact that there is very little

effort to be made for the thruster in order to keep the yaw angle at a constant value. The control effort produced by the PID-controller takes on values in a small range around zero, which in turn forces the thruster to reverse it's rotational direction very often. The thruster motor, being a brushless DC-motor does not exhibit great preformance when the rotational direction is rapidly changed. However, the oscilations are fairly small in magnitude, and can barely be observed in the images captured by the camera.

In terms of speed, the controller fairs very well. Desired angle of 30 degrees is reached within about 1.5 seconds. If the camera system was ever to be used for tracking purposes of a rapidly moving object, the controller is likely to achieve high agility.

The controller does suffer from a small steady state error which is highly present in the left part of figure 5.1, where the measured yaw angle is at a steady 4 degrees.

5.1.2 Trajectory tracking

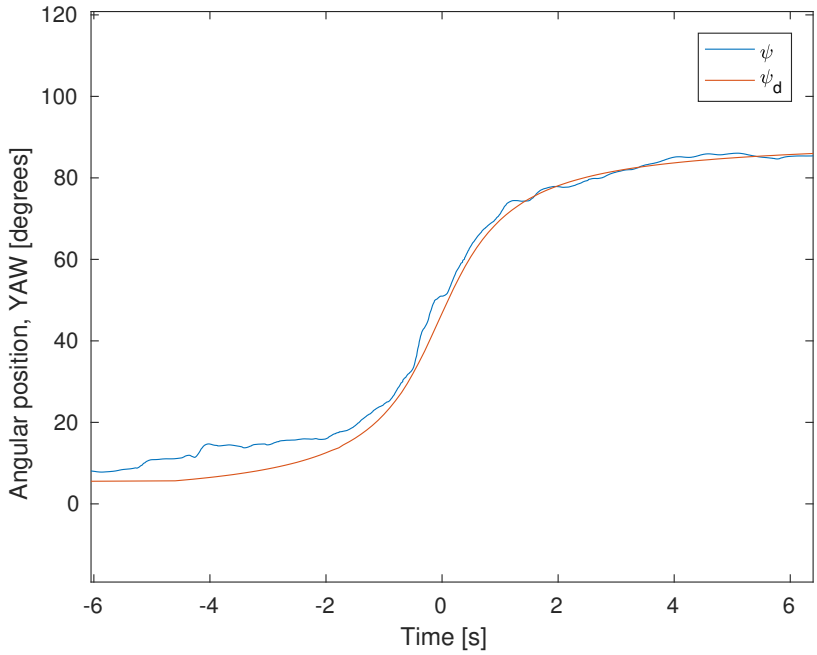


Figure 5.2: Trajectory tracking, wet lab. $\psi_d = \frac{2}{\pi}45arctan(1.1t) + 45$

In this test, the desired yaw angle ψ_d is always moving, which in turn gives the thruster a lot more work to do. As a result, the small oscillations are significantly reduced and the controller seems to be working even better in regions where the trajectory is at a high rate of change.

5.2 Field experiments

5.2.1 Step response

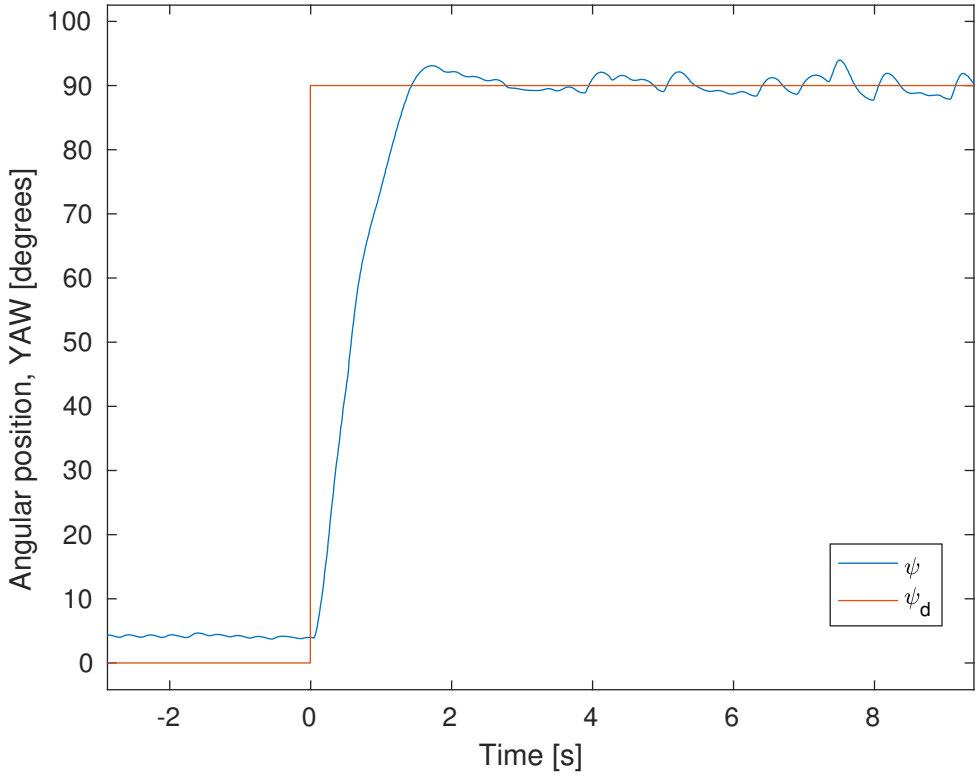


Figure 5.3: Step response, field test.

The step response test in the field shows that the controller is having a hard time setteling at 90 degrees. Although there were small oscillations in the lab test, they seem to have increased when exposed to the real world sea conditions. Once again, this behaviour accours when the rotational direction of the motor is rapidly changed. Another factor to be aware of is the fact that the parameters of the PID-controller is tuned in the wet lab. Better results may be possible if the controller is tuned in the field.

5.2.2 Trajectory tracking

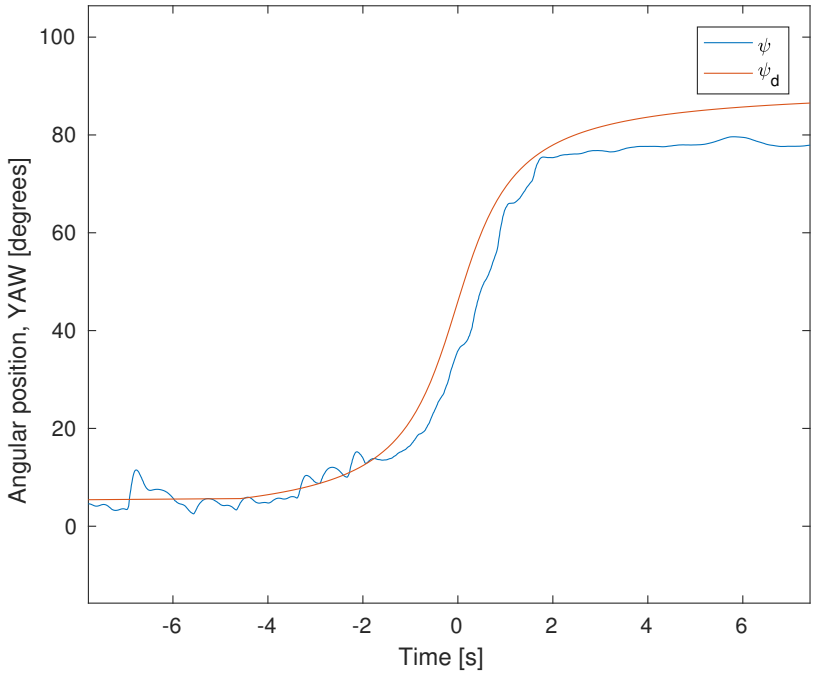


Figure 5.4: Trajectory tracking, field test. $\psi_d = \frac{2}{\pi}45arctan(1.1t) + 45$

Once again, the result from the field test shows a decrease in preformance compared to the lab test, which is expected. Nontheless, the controller is able to follow the trajectory to a certain degree.

5.2.3 Vertical free fall

During the free fall test, raw data from the 3 axis accelerometer was logged, which is plotted in figure 5.5. This plot is a little to read, so a second plot of the combined magintude is presented in figure 5.6. The magnitude of the accelereation is defined as

$$|a| = sqrtx^2 + y^2 + z^2 \tag{5.1}$$

where $|a|$ is the acceleration magnitude, and x, y, z are acceleration measurements in their respective direction.

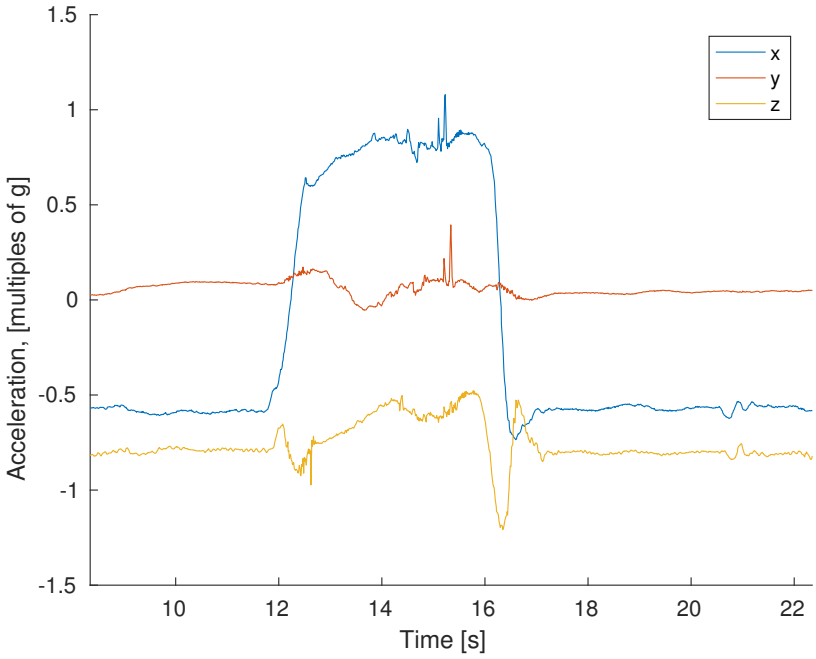


Figure 5.5: Vertical free fall acceleration in x- y- and z-direction.

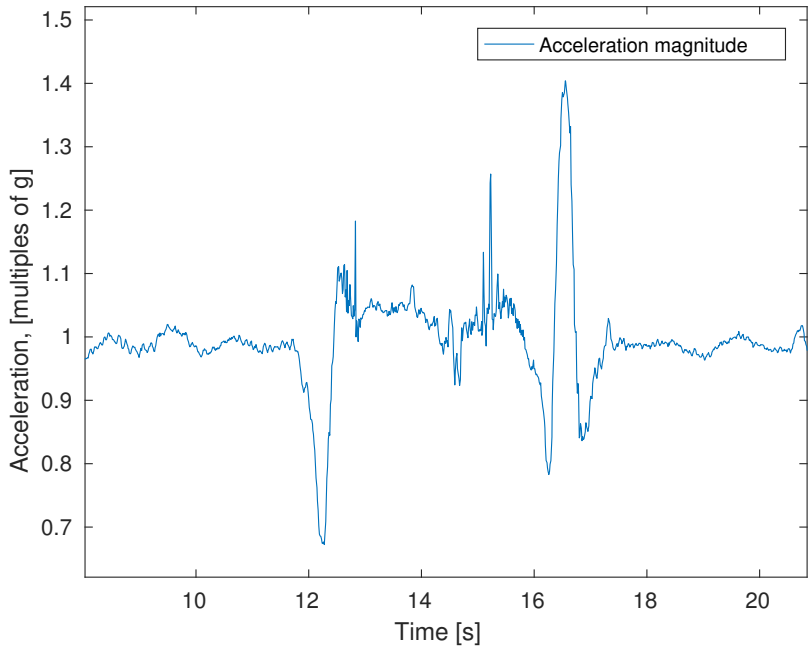


Figure 5.6: Vertical free fall acceleration magnitude.

Initially, the camera is kept still just below the surface of the water, measuring in at 1g $(9.1\frac{m}{s^2})$ due to the rope. The negative spike in figure 5.6 appears when the rope holding up the camera system is let go and the system is left in a state of free fall. The positive spike appears when the camera system is stopped by the rope reaching it's maximum length of three meters.

The interesting part about this plot is the time inbetween the spikes, indicating how long it takes for the system to drop three meters in a free fall situation. This time frame can be used as a guideline when the drag coefficients of the equations of motions are determined.

5.3 Pitch controller

Installing a brushless DC-motor for the pitch actuation, turned out to be a bad choice. As there is no device in the motor for measuring the rotors rotational position relative to the stator, the ESC must rely on an estimate of the position based on the the backemf induced back to the ESC as the rotor rotates. This principle works well as long as the motor maintains a certain speed, but as soon as the motor is brought to a halt, the position can no longer be estimated. When the motor is starting from still stand, it is running in open loop, that is, without the knowledge of the rotor position.

Unfortunately, the time frame for this project did not allow for the replacement of

the motor, leaving the pitch unactuated until the innstalation of a new motor.

Chapter 6

Conclusion

An underwater camera system with motion control in pitch and yaw has been developed, implemented in full scale, and tested. The water jet propulsion system for the yaw control shows promising results even though it was somewhat overdimensioned for this particular application. The PID-controller yields good results whenever the circumstances allows the control effort to be in a range that does not include zero. In other words, when the motor is constantly running in the same direction.

The pitch actuation on the other hand, gave very little results other than to confirm that a brushless DC-motor works very poorly in low speed angular positioning.

The implemented camera system and it's physical configuration is an interesting approach for motion control in an underwater environment. It has the advantages of being fairly lightweight, quickly deployable and easy to modify. A great deal of potential still remains to be explored.

Chapter 7

Further Work

7.1 Modifications

The brushless DC-motor currently installed in the pitch actuator needs to be replaced by a motor better suited for low speed, high precision angle control. In addition, the camera house would benifit from a weigth distribution analysis so that the center of bouyancy and center of mass coincides, which in turn would enhance the balance in the pitch actuator.

The water jet propulsion system is somewhat overdimensioned, both in terms of size and thrust. A smaller tube, and a smaller thruster is likely to yield better results.

7.2 Implementing object tracking

One of the priorities while developing the camera system was to keep in mind future work that goes beyond the scope of this thesis. The Jetson TX1 is highly capable of running machine vision algorithms alongside the PID-controllers, which for example can be used for an object tracking scheme.

7.3 Model equations

The model equations presented in this thesis still needs to be further developed. It would be very interesting to se if a model based controller yields better results than the PID controllers currently implemented.

Bibliography

[1] National oceanic and atmospheric administration. http://www.nmfs.noaa.gov/aquaculture/faqs/faq_aq_101.html. Accesed: 05.06.2017.

[2] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[3] SNAME. Nomenclature for treating the motion of a submerged body through a fluid. *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin No.*, pages 1–5, 1950.

[4] Sony. Camera specification fcb-ev7520. http://www.image-sensing-solutions.eu/FCB-EV7520.html, 2017. Accessed: 2017-05-08.

[5] Seed studio product info. http://wiki.seeedstudio.com/wiki/Grove_-_IMU_9DOF_v1.0.

[6] Connecttech product info. http://connecttech.com/product/orbitty-carrier-for-nvidia-jetson-tx2-tx1/. Accesed 08.06.2017.

[7] Blue robotics product info. http://bluerobotics.com/store/thrusters/t200-thruster/. Acceced: 08.06.2017.

[8] Matek systems production information. http://www.mateksys.com/?portfolio=hub5v12v.

[9] A versatile 9-dof imu library for embedded linux systems. https://github.com/mrbichel/RTIMULib.

[10] Andy Zelenak. http://wiki.ros.org/pid.

Appendix

The contents of the digital appendix is arranged as follows.

ROS Software

Matlab tools

Simulation tool

ROS plotting tool

Solidworks models

Demo video