

Design and verification of a stabilizing control system for maritime searchlights

Master's thesis in Systems, Control and Mechatronics

JACOB ANDRÉN & JOAKIM BOODH

MASTER'S THESIS 2019

Design and verification of a stabilizing control system for maritime searchlights

Roll, pitch and heave stabilization based on IMU data, mathematical modeling and control implementation.

Jacob Andrén & Joakim Boodh



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

© JACOB ANDRÉN & JOAKIM BOODH, 2019.

Supervisor: Balázs Kulcsár, Department of Electrical Engineering
Advisor: Jonas Williamsson, CONSAT AB
Examiner: Balázs Kulcsár, Department of Electrical Engineering

Master's Thesis 2019
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in L^AT_EX
Gothenburg, Sweden 2019

Design and verification of a stabilizing control system for maritime searchlights
Jacob Andrén and Joakim Boodh
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Maritime vessels are generally equipped with several outdoor lighting fixtures, one being the searchlight. The searchlight play a key role during navigation at night and during search and rescue operations in dark conditions. Most, if not all, searchlights on the market today have the option to control the tilt and pan of the light. However, the maneuvering of such a light becomes increasingly difficult when the conditions of the sea becomes rough. Hence, a stabilization of the light beam could help facilitate the searchlight operator's control of the light. In this thesis, a control system for searchlights exposed to movements in roll, pitch and heave was developed by controlling the lights pan and tilt. Data from an *IMU* sensor measuring the boats movement were used to track a calculated reference point by the development of a mathematical algorithm. A mathematical description of the light was then derived to form the basis for the design of three controllers; *PID*, *LQR* and *LQI*. To verify the control system simulations were performed and an implementation on a physical prototype were made. The results proved that the developed control system would stabilize the beam of a searchlight based on *IMU* data and by feeding back the positions of the searchlight's motors.

Keywords: Control, Control Systems, Stabilization, Maritime applications, searchlight, Mathematical modeling, IMU, Roll, Pitch, Heave, PID, LQR, LQI, Simulations, Physical Testing

Acknowledgements

We would like to give a sincere thank you to *Balázs Adam Kulcsár* for supervising this project with an never ending enthusiasm and supporting mood. A big thanks also goes out to *Jonas Williamsson* for making this whole project possible and doing all he can to fulfill the projects every need. We would also like to thank *Stefan Persson* and *Anders Ödman* from the *Norwegian Society for Sea Rescue* for being both courteous and helpful when ever needed. A last thank you is given to *Jens Wilhelmsson* for keeping spirits up all through the project.

Jacob and Joakim, Gothenburg, January 2019

Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Background	1
1.1.1 Similar studies	2
1.1.2 CLITE 2	2
1.2 Objective	3
2 Prologue	5
3 Theory	7
3.1 Reference frames	7
3.2 Transformation matrices	8
3.3 Control system	8
3.3.1 Physical modelling	9
3.3.2 PID	10
3.3.3 LQR	11
3.3.4 LQI	12
4 Method	15
4.1 Mathematical modelling	15
4.1.1 Introduction to concept choice	15
4.1.2 Tracking a reference point	16
4.1.2.1 Reference point calculation	16
4.1.2.2 Light angle calculations	17
4.2 Control system design	20
4.2.1 State space representation	20
4.2.2 Controllers	22
4.2.2.1 PID	23
4.2.2.2 LQR	23
4.2.2.3 LQI	25
4.3 Simulations	26
4.4 Physical testing	26
4.4.1 Test equipment	26
4.4.2 Test procedure	28

5	Results	31
5.1	Simulation	31
5.2	Physical testing	33
6	Discussion	37
6.1	Mathematical model	37
6.2	Simulation results	37
6.3	Physical testing	38
6.3.1	Hardware	38
6.3.2	Tuning of controllers	39
6.3.3	Testmethod	40
6.4	Implementation on the <i>CLITE 2</i>	40
7	Conclusion	41
	Bibliography	43
A	Appendix	I
A.1	Simulink model for simulation of the system	II
A.2	Simulink model subsystem for calculation of reference angles	III
A.3	Hardware used for test and verification	IV
A.4	Determining light parameters	IV

List of Figures

1.1	<i>CLITE 2</i> , a searchlight developed and sold by the company <i>Luminell</i> . Photo by <i>Luminell</i>	2
3.1	The boat with the complete set of reference frames with its necessary angles.	7
3.2	Circuit diagram of an electric motor	9
3.3	Overview of the constants and variables used to calculate the me- chanical dynamics of the system.	10
3.4	Block diagram showing an implementation of a PID controller.	11
3.5	Overview of a closed loop system controlled by an LQR with feedback gain K	12
3.6	Overview of a closed loop system controlled by an <i>LQI</i> with feedback gain K_i	13
4.1	Illustration of a boat tracking a reference.	16
4.2	Orthogonal projection of reference coordinate onto the xy -plane of the l -frame. The xy -plane is colored orange	18
4.3	Light rotated γ , pointing towards Ref^{xy}	19
4.4	Light rotated its calculated tilt angle η	20
4.5	Zeros (o) and poles (x) of the closed loop system using <i>PID</i> controller.	23
4.6	Zeros (o) and poles (x) of the closed loop system using <i>LQR</i> controller.	24
4.7	Zeros (o) and poles (x) of the closed loop system using <i>LQI</i> controller.	25
4.8	Simplified overview of the simulation setup as implemented in <i>Simulink</i>	26
4.9	Test rig used to emulate the two engines in the <i>CLITE 2</i>	27
4.10	Overview of the implemented hardware	27
4.11	The test rig placed on a <i>6DOF</i> platform for testing	28
4.12	The movement of the <i>6DOF</i> platform when testing the controllers.	28
4.13	Sketch of the test setup used during physical testing.	29
5.1	Reference which the searchlight is set to point at with a new reference being set after 5 seconds.	31
5.2	Boat motion and the resulting light angles calculated by the system based on the reference coordinate in figure 5.1.	31
5.3	The three controllers response to a step input.	32
5.4	A close up of the three controllers response to a step input.	32
5.5	The controllers ability to track moving reference angles as a result of the boats motion.	32

5.6	The squared error between the pan angle reference and controller output.	33
5.7	The squared error between the tilt angle reference and controller output.	33
5.8	The reference pan angle compared with the measured angle for three different controllers during physical testing.	34
5.9	The reference tilt angle compared with the measured angle for three different controllers during physical testing.	34
5.10	Light <i>POI</i> for the system without any controller.	35
5.11	Light <i>POI</i> for the system with PID controller.	35
5.12	Light <i>POI</i> for the system with <i>LQR</i> controller.	35
5.13	Light <i>POI</i> for the system with <i>LQI</i> controller.	35

List of Tables

4.1	Light parameters used to verify system	21
5.1	MSE values for the error between the reference and the controller output.	33
5.2	MSE values for the error between the reference and the controller output.	35

Nomenclature

Below is an overview of terms and abbreviations that appear in the thesis

6DOF: <i>6 Degrees of freedom.</i> A platform with the ability to move in six directions.	e_{EMF} : Back-EMF measured in voltage.
A: State space system matrix	η : <i>Eta.</i> Tilt angle of light.
A_a : Augmented state space system matrix	FIR: <i>Finite impulse response.</i> A filter whose impulse response is finite. Often used to smooth out a signal.
A_d : Discrete time state space system matrix.	$G(z)$: Transfer function.
ADRC: <i>Active disturbance rejection control.</i> Controller that inherits from <i>PID</i> with an added state variable that aims to reject disturbances.	γ : <i>Gamma.</i> Pan angle of the light beam.
α : <i>Alpha.</i> Euler angle around x -axis.	H: <i>Henry.</i> Inductance unit.
b : Damping coefficient.	Heave: Translation in z direction.
B: State space input matrix.	h -frame: Frame with respect to the hydrodynamics of the boat.
B_a : Augmented state space input matrix.	I: Identity matrix.
B_d : Discrete time state space input matrix.	i : Current in a circuit.
β : <i>Beta.</i> Euler angle around y -axis.	IMU: <i>Inertial measurement unit.</i> Sensor that measures a body's angular rate and accelerations.
b -frame: Frame with respect to the boat.	J: Moment of inertia.
c_1 : Scaling x in xy -plane of l-frame.	J_c : Cost function.
c_2 : Scaling y in xy -plane of l-frame.	K: Feedback gain for <i>LQR</i> .
c_3 : Scaling z in xy -plane of l-frame.	kg : <i>Kilogram.</i> Mass unit.
c_4 : A constant.	K_i : Feedback gain for <i>LQI</i> .
C: State space output matrix.	K_m : Constant relating torque with input current.
C_a : Augmented state space output matrix.	K_u : Constant relating rotational velocity to back-EMF.
C_d : Discrete time state space output matrix.	L: Capacitance.
D: State space feedforward matrix.	L_{beam} : Length of light beam.
d_l : Vector indicating distance between light and b-frame.	l -frame: Frame with respect to the light.
d_k : Distance of light to b-frame, where $k = x, y$ and z	LTI: <i>Linear time invariance.</i> A system is said to be <i>LTI</i> if the system is linear and the output, given an input, yields the same result independent of time.
D_d : Discrete time state space feedforward matrix.	LQI: <i>Linear-Quadratic-Integral.</i> Controller similar to an <i>LQR</i> but with an integral action.
e : Error between reference and controller output.	LQR: <i>Linear-Quadratic-Regulator.</i> Controller operating on linear

system.	Ref_k^p : Reference in coordinate k with respect to p .
m^2 : <i>Square meters</i> . Area in meters.	R_X : Euler rotation matrix around X , also denoted roll.
MPC : <i>Model predictive control</i> . Control method which fulfills a set of constraints.	R_Y : Euler rotation matrix around Y , also denoted pitch.
MSE : <i>Mean squared error</i> . Measures the average of the squared error of a signal.	R_Z : Euler rotation matrix around Z , also denoted yaw.
Nm : <i>Newton meter</i> . Torque measurement.	s : <i>Seconds</i> . Unit of time.
Ω : <i>Ohm</i> . Resistance unit.	<i>Surge</i> : Translation in x direction.
ω : <i>Omega</i> . Angular velocity.	<i>Sway</i> : Translation in y direction.
P : Unknown variable in Ricatti equation.	T : The torque of a shaft.
Pan : The light's angle around its z axis.	Θ : <i>Theta</i> . Angle of motor shaft.
PID : <i>Proportional-Integral-Derivative</i> . Controller consisting of three parts; a proportional, an integral and a derivative part.	<i>Tilt</i> : The light's angle around its y axis.
<i>Pitch</i> : Rotation around the y -axis.	T_k^p : Transformation matrix from k -frame to p -frame.
P_k^p : Position of p -frame origin expressed in k .	T_s : Sample time.
POI : <i>Point of illumination</i> . The point where the searchlight's beam ends.	u : Control input signal.
ψ : <i>Psi</i> . Euler angle around z -axis.	u_a : Augmented control input vector.
Q_u : Weight matrix for input in the cost function J .	v : Input voltage.
Q_x : Weight matrix for state vector in the cost function J .	V : <i>Voltage</i> . Unit for electrical potential.
R_m : Reachability matrix.	W : <i>Watt</i> . Unit of power.
R : Electric resistance.	w : Heave motion of a boat.
r : Reference signal.	x : State vector.
R_a : Augmented reachability matrix.	x_1 : First element in state vector.
RC : <i>Resistor capacitor</i> . Short for the two components resistor and capacitor, mounted in parallel to form an RC -filter.	x_2 : Second element in state vector.
<i>Roll</i> : Rotation around the x -axis.	x_3 : Third element in state vector.
RPM : <i>Revolutions-per-minute</i> . Angular velocity.	x_a : Augmented state vector.
	x_b : x axis in b-frame.
	x_h : x axis in h-frame.
	x_l : x axis in l-frame.
	y : State space output.
	<i>Yaw</i> : Rotation around the z -axis
	y_b : y axis in b-frame.
	y_h : y axis in h-frame.
	y_l : y axis in l-frame.
	z : A complex variable.
	z : Integral state in LQI .
	z_b : z axis in b-frame.
	z_h : z axis in h-frame.
	z_l : z axis in l-frame.

1

Introduction

On board offshore and maritime vessels, such as oil tankers and rescue ships, the need for outdoor lighting fixtures is oftentimes essential. These fixtures are usually located on parts, or all around, the ship in order to light up the deck, or the area surrounding the vessel. In addition, these lights play a key role during navigation at night in order to find possible obstacles in the sea such as breakers and icebergs as well as during search and rescue operations. Several companies has realized the need for these lights and producers such as *DHR*, *Techno Marine Group*, *Hella Marine* and *Luminell* are supplying the demand. As a result a vast range of products are available on the market today with lights that differ both in type, design and size depending on their area of application, one being the searchlight.

Searchlights have a narrow yet strong beam in order to light up specific areas. Most, if not all, searchlights on the market today have the option to control its tilt and pan direction, increasing the lights flexibility significantly. However, the maneuvering of such a light becomes increasingly difficult when the conditions of the ocean becomes rough. In a situation where a boat is rocking considerably due to waves, it disrupts the maneuvering. The searchlight operator have to control the light beam constantly as the boat rocks back and forth. This poses as a big issue if the light, for instance, is being used to find a person that has fallen over board. If the operator could avoid compensating for these movements, the maneuvering would hopefully be made easier, and the risk of, for instance, loosing sight of a target reduced.

To investigate the possible benefits with searchlight stabilization the company *Luminell* sought to develop a control system for their products. This idea came as a part of the development of a new type of searchlight the company was developing during the year of 2018. This thesis was being done as a part of this development with the goal of providing a basis for a control system able to be implemented on *Luminell's* existing searchlight *CLITE 2* [1], with the hope of possible implementation on future products as well.

1.1 Background

Control systems for maritime purposes exists today, but not for searchlight applications. In this section similar systems will be presented as well as an introduction to the light sought to be controlled.

1.1.1 Similar studies

As of today, none of the previously mentioned manufacturers have a solution for aiding the operator during rough seas, i.e. stabilizing a searchlight. However, similar solutions have been constructed for other purposes. In one research paper stabilization of a camera mounted on floating supports was investigated [2]. The researchers used an inertial measurement unit (*IMU*) together with a proportional–integral–derivative controller (*PID*) to control two stepper motors steering the cameras pan and tilt direction. The system was experimentally tried on a six degrees of freedom (*6DOF*) pneumatic hexapod and showed that the proposed solution was able to somewhat reject disturbances in the form of pitch, roll and yaw motions, resulting in a stabilized camera image.

Within the military, vast amounts of research have been done regarding stabilization. Limiting the scope to systems subjected to non-linear movements, such as gun stabilization on tanks and gunships, a span of different control methods have been put to the test. With examples as, Model Predictive Control (*MPC*) [3], Sliding Mode Variable Structure Control [4], Active Disturbance Rejection Control (*ADRC*) [5] and the results vary but with generally positive results.

1.1.2 CLITE 2

CLITE 2, seen in figure 1.1, is a searchlight aimed for small to mid-sized boats in the range of up to 50 *m* in length and are often found on yachts, patrol-, SAR-, pilot-, and workboats [1]. The light is controllable by a joystick in two directions; pan and tilt. The rotation is made possible by the help of two brushless DC-motors, one for every degree of freedom, with possible rotation of 360°. The motors are equipped with rotary encoders in order to read their positions.



Figure 1.1: *CLITE 2*, a searchlight developed and sold by the company *Luminell*. Photo by *Luminell*.

The light is usually mounted on a high point on the water vessel, i.e. the roof or in a mast. With its two LED light sources of 250 watts each the light has a range of

1500 m [1].

1.2 Objective

This master's thesis aims to develop a control system that stabilizes the beam of a searchlight when subjected to disturbances in the form of waves causing the boat to pitch, roll and heave. The control system will be designed for the searchlight named *CLITE 2* produced and sold by the company *Luminell*. As part of the development, different types of controllers will be implemented and tested in order to investigate the performance of the concept.

2

Prologue

With the basis of the project being to develop a system that would control a searchlight, and more specifically reject disturbances in the form of waves, different approaches could have been taken in order to adequately solve the problem. From the company's point of view, the expectation was to be able to keep the light steady when the boat was pitching. This could have been done in several ways. The most simple would have been to mount the light in some form of gimbal [6]. This would have kept the light itself steady when subjected to waves, but it would not have considered where the light shined at. One solution to solve this could have been to take advantage of that the *CLITE 2* also could come equipped with a thermal camera. Using this, some form of image recognition together with an object tracking algorithm could have been implemented in order to track an object in the water and keep the light focused at a point. Implementing this solution would not only ensure stabilization during pitching, but also if the boat would be subjected to roll, yaw, surge, sway and heave. This solution would however be dependent on that very thermal camera, which would limit the possibility to implement the solution on searchlights without a camera, and also increase the cost for a possible end consumer, hence it was not an option.

In addition to the company's requirements, the thoughts of an end user¹ was taken into consideration. It was understood that the purpose of a disturbance rejecting searchlight would be to keep the light beam stable, in the eyes of the operator. The light would in other words not need to track an object, but it needed to track a point with respect to the boat. That meant that the point would keep its relative position to the boat, even though the boat was moving forward or turning. With this in mind it was decided that the control system would handle pitch, roll and heave motions. To do this, it was essential to know the change of orientation of the boat in order to compensate for it. After having a dialogue with the company it became clear that they were planning on installing an *IMU* sensor in their next generation searchlight. Knowing this, an *IMU* was chosen, being a cheap and accessible sensor that would fulfill the demand.

To be able to control the light a controller had to be chosen. As described in the beginning of chapter 1, several types of controllers have been tried for concepts similar to controlling a searchlight. Since this projects aim was to also provide something that could become a working product in the end, less experimental controllers were

¹Stefan Persson, a former Norwegian coast guard employee, helped share his experiences of using searchlights during the years.

needed in order to guarantee its reliability. With that said, three types of controllers were chosen to be tested on the light; PID, Linear-Quadratic-Regulator (LQR) and Linear-Quadratic-Integral controller (LQI). It was also chosen to use the motor encoders to give position feedback to the controllers.

3

Theory

In this chapter necessary theory such as definitions, mathematics, dynamics and controllers will be presented.

3.1 Reference frames

To accurately describe the motion and orientation of a ship the definition of reference frames becomes imperative. A ship has six degrees of freedom, three being translations and three being rotations around each axis. To sufficiently describe a ship's orientation in the purpose of this thesis, the reference frames presented in figure 3.1 were necessary. The frames and notations seen in the figure will be used for the rest of this thesis. In this section a brief explanation of each frame will follow.

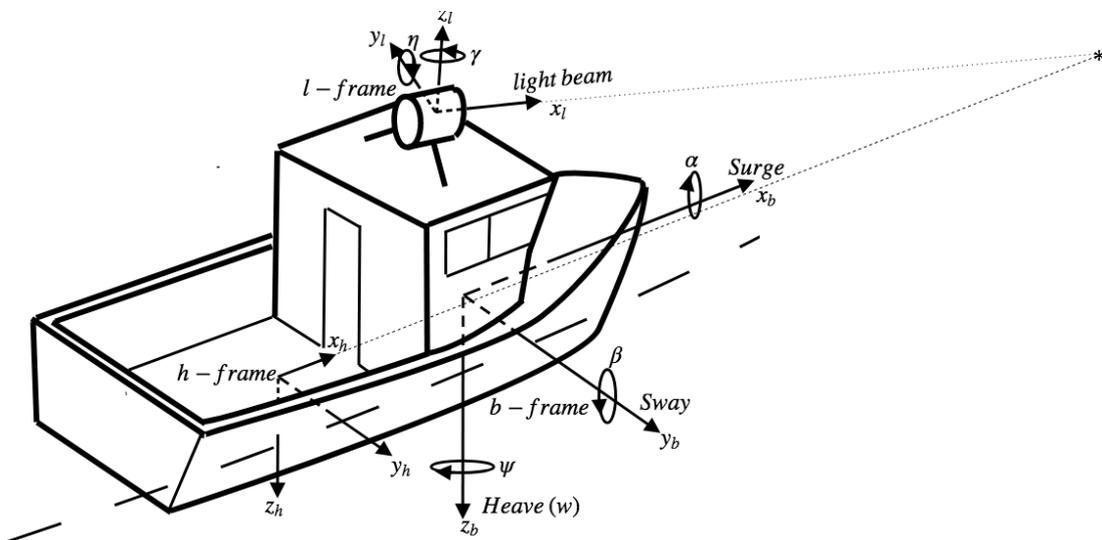


Figure 3.1: The boat with the complete set of reference frames with its necessary angles.

Hydrodynamic frame (*h-frame*)

The hydrodynamic frame follows the direction of the boat but is vertically attached to the mean of the water surface. That means that its origin follows the mean speed of the ship and that x_h is pointing forward, y_h to starboard and z_h downwards.

Body-fixed frame (*b-frame*)

The origin of the *b-frame* is also attached to the hull but coincides with the principal axes of inertia. It is considered an inertial frame since it rotates and moves with respect to the *h-frame*.

Light frame (*l-frame*)

The searchlight is mounted to the boat at a fixed distance from the origin of the *b-frame*. However, the orientation of the *l-frame* is decided by its own degrees of freedom, pan and tilt, the rotations around its *y*- and *z*-axes respectively and are denoted η and γ .

3.2 Transformation matrices

In order to be able to relate the pose of the frames in section 3.1 with respect to each other, rotation and transformation matrices were used. The rotation matrices are based on Euler (or roll-pitch-yaw) angles, where rotations in 3D-space are described as three consecutive rotations around its coordinate axis[7]. The Euler angles around each axis *x*, *y* and *z* are denoted α , β and γ respectively, as seen in figure 3.1, and their rotation matrices are described as follows

$$\begin{aligned}
 R_X(\alpha) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{bmatrix} \\
 R_Y(\beta) &= \begin{bmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{bmatrix} \\
 R_Z(\psi) &= \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.1}$$

In the report, $\cos()$ will be denoted $c()$ and $\sin()$ will be denoted $s()$. It is convenient to include the origin of each frame when relating or moving between them. This is achieved by interpreting a transformation matrix, T , which incorporates the coordinates of the frame origin, here denoted P , in the same matrix as the rotation. The transformation matrix is defined as

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix} \tag{3.2}$$

3.3 Control system

The control system developed in this thesis was decided to be based on three different controllers; *PID*, *LQR* and *LQI*. In order for the controllers to perform well,

the real system required an accurate mathematical description. In this section a mathematical description of the system will be made, followed by descriptions of the three controllers and basic theory behind them.

3.3.1 Physical modelling

The light was divided into two subsystems; one electrical and one mechanical. The electrical part consisted of two electric motors, one for each degree of freedom. Since the two motors were assumed to be the same, both of them could be described with the same equations. In figure (3.2) a representation of the components that make up an electric motor is illustrated.

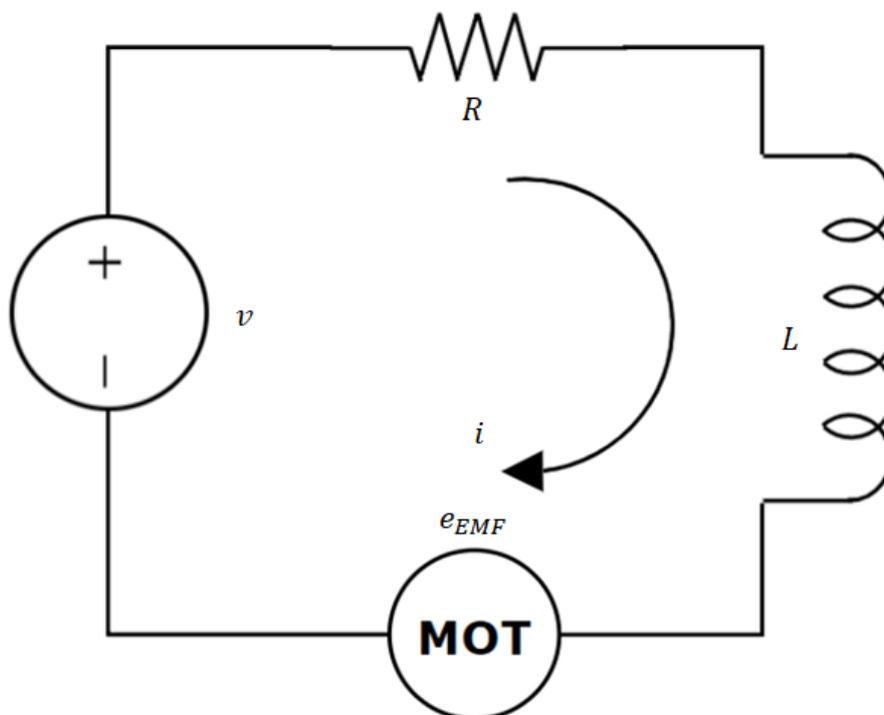


Figure 3.2: Circuit diagram of an electric motor

In accordance with Kirchhoff's law [8] the first order equation describing the electrical circuit seen in figure 3.2 was derived as

$$v - Ri - \frac{di}{dt}L - e_{EMF} = 0 \quad (3.3)$$

where v is the input voltage, R the resistance, L the inductance, e the back-EMF and i the current in the circuit. The back-EMF is also proportional to the rotational velocity as

$$e_{EMF} = K_u\omega \quad (3.4)$$

where K_u is a constant relating the rotational velocity to the induced back-EMF.

The mechanical part of the system, i.e the moving parts of the light such as drive shafts and light housing, were described as a spinning axle as shown in figure 3.3

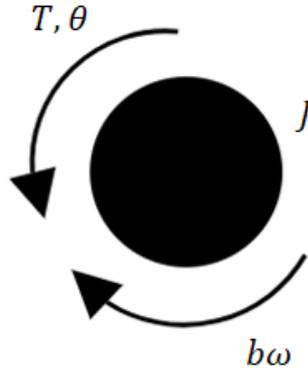


Figure 3.3: Overview of the constants and variables used to calculate the mechanical dynamics of the system.

where T is the torque, θ the angle, J the moment of inertia, b the friction coefficient and ω the rotational velocity¹.

According to Newton's second law of rotation [9], the system illustrated in figure 3.3 could be derived as the second order equation

$$J\dot{\omega} = T - b\omega \quad (3.5)$$

where the motor torque is proportional to the current with the constant K_m relating the proportionality

$$T = K_m i \quad (3.6)$$

3.3.2 PID

A *PID* is a feedback controller that is commonly used in a variety of control loop systems [10]. As the name indicates, it consists of three parts; a proportional constant, an integrating- and a derivation part. Each consist of a design parameter that can be tuned to achieve desired performance. An implementation of the controller is shown in figure 3.4 and the controllers mathematical description [11] is

$$u = Pe + I \int_0^t e(\tau) d\tau + D \frac{de}{dt}, \quad (3.7)$$

where P, I and D are the design parameters and e is the error between the reference and the measured output.

¹Note that $\theta = \int \omega dt$

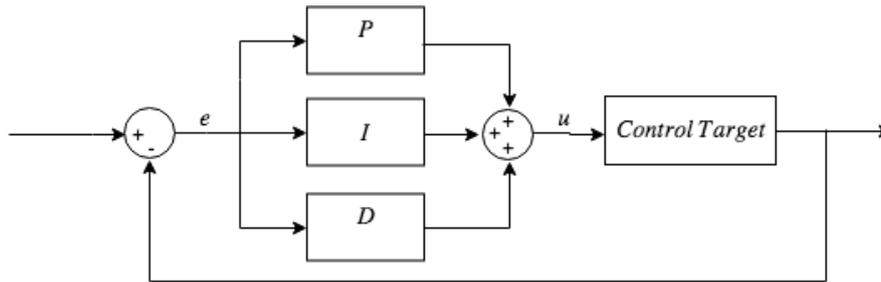


Figure 3.4: Block diagram showing an implementation of a PID controller.

3.3.3 LQR

An *LQR* is an optimal controller that operates on linear systems expressed on state space form as

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (3.8)$$

The *LQR* is considered optimal since it minimizes the cost function J_c [12], defined as

$$J_c = \int_0^{\infty} (x^T Q_x x + u^T Q_u u) dt \quad (3.9)$$

where x is the state vector, u is the control signal, Q_x and Q_u are design parameters.

The design parameters penalize either the state vector or the input vector. That means that high values in Q_u with respect to $x^T Q_x x$ penalizes the input vector, trying to stabilize the system with less energy. This approach is called expensive control strategy while low values of Q_u is said to be cheap control strategy. The values of Q_x are also designed whether changes in states are to be allowed easily or not.

A requirement for implementing an *LQR* is that the system to be controlled is reachable [11], meaning that the matrix R_m in equation (3.10) is full rank. The system can also be controllable, meaning that the *Rank* R_m equals the *Rank* A . R_m is defined as

$$R_m = \begin{bmatrix} A & AB & A^2 B \dots A^{n-1} B \end{bmatrix} \quad (3.10)$$

where n is the number of states in the state vector.

Given that the system is reachable, the optimal feedback gain, K , seen in figure 3.5, can be calculated in accordance with equation (3.12). In order to do so, the unknown variable P needs to be defined with the algebraic Ricatti equation, shown in equation (3.11).

$$A^T P A - P - A^T P B (Q_u + B^T P B)^{-1} B^T P A + Q_x = 0 \quad (3.11)$$

$$K = -(B^T P B + Q_u)^{-1} B^T P A \quad (3.12)$$

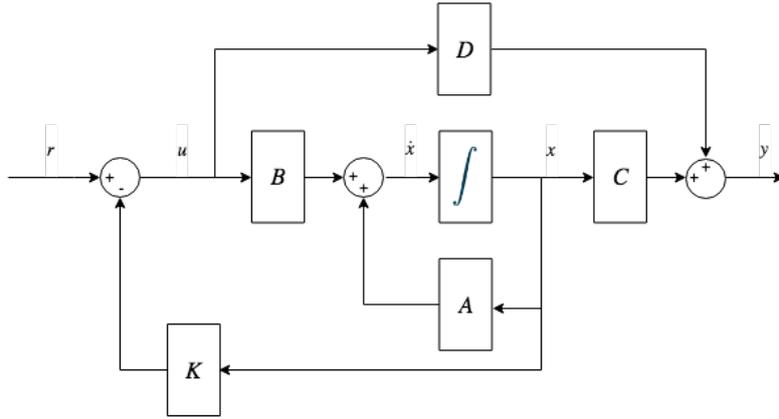


Figure 3.5: Overview of a closed loop system controlled by an LQR with feedback gain K .

An evaluation of the block diagram in figure 3.5 reveals the closed loop system equations.

$$\begin{aligned} \dot{x} &= (A - BK)x + Br \\ y &= (C - DK)x + Dr \end{aligned} \quad (3.13)$$

where r is the reference and the control input signal is now $u = -Kx + r$.

3.3.4 LQI

The *LQI* is an *LQR* with an added integral action [12]. With the *LQI*, the state vector is extended with an integral state, denoted z , and defined according to

$$z = \int y(t) - r(t) dt \quad (3.14)$$

where $y(t)$ is the output signal and $r(t)$ the reference.

To include z into the state space it must be derived with respect to time, as follows

$$\dot{z} = y(t) - r(t) = Cx(t) + Du(t) - r(t) \quad (3.15)$$

The integral state is incorporated in an augmented state vector, subscripted a , which in turn requires augmented state space matrices in order to satisfy equation (3.15). This in turn leads to a new state space representation as follows

$$\begin{aligned} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix}}_{\dot{x}_a} &= \underbrace{\begin{bmatrix} A & 0_{n \times q} \\ C & 0_q \end{bmatrix}}_{A_a} \underbrace{\begin{bmatrix} x \\ z \end{bmatrix}}_{x_a} + \underbrace{\begin{bmatrix} B & 0_{n \times q} \\ D & -I_q \end{bmatrix}}_{B_a} \underbrace{\begin{bmatrix} u \\ r \end{bmatrix}}_{u_a} \\ y &= \underbrace{\begin{bmatrix} C & 0_q \end{bmatrix}}_{C_a} \begin{bmatrix} x \\ z \end{bmatrix} + Du \end{aligned} \quad (3.16)$$

where n is the number of states and q is the number of integrators. Note that the reference, $r(t)$, now is considered as an input to the system. The augmented matrices, subscripted a in equation (3.16), are used to calculate the feedback gain K_i

4

Method

This project could be summed up to three separate stages: mathematical modelling, simulations and physical testing. What these entailed and what was performed will be explained in this chapter.

4.1 Mathematical modelling

As mentioned in chapter 2, the aim was to track a point with respect to the boat. In order to mathematically describe the tracking of a point while a boat is moving, several equations had to be derived. To get a better understanding of what tracking a reference point would entail, this section will firstly introduce an example.

4.1.1 Introduction to concept choice

Mentioned in chapter 2, it was decided to track a reference relative the boat, more specifically track a reference with respect to the *h-frame*. To give a better understanding of the basic principle of having a searchlight tracking a reference coordinate, a simple case is presented.

In figure 4.1, an example was drawn. In the figure, a boat is standing upright, with a searchlight pitching down with an angle, η , pointing in the same direction as that of the boat. The beam of the light penetrates the surface of the water in front of the boat. This point will hereafter be denoted as the *point of illumination (POI)*. When the system is initiated, the *POI* will be set as the reference coordinate with respect to the *h-frame*, denoted $Ref = [Ref_x \ Ref_y \ Ref_z]^T$. As long as an operator does not change the attitude of the light, the *POI* shall be controlled to stay at the *Ref*. In this example, the reference to track would be $[Ref_x, 0, 0]^T$.

In the case of the boat pitching with an angle, β , the lights pitch angle, η , would need to adjust in order to keep the *POI* positioned on the reference point, *Ref*. By doing so, the light will track the reference point, no matter the movement of the boat.

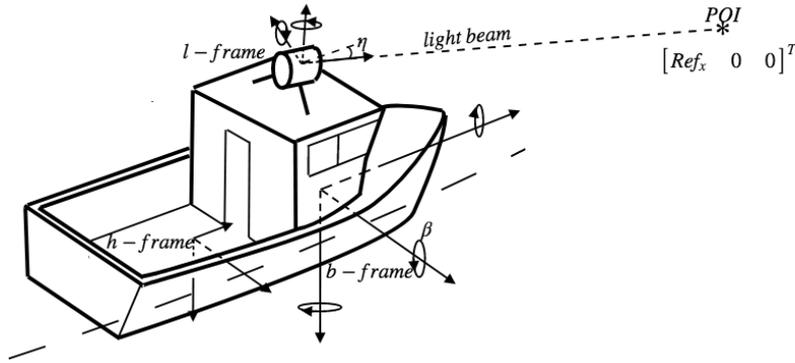


Figure 4.1: Illustration of a boat tracking a reference.

In the following sections, a detailed description of the implemented methods for calculating and tracking a reference coordinate with a searchlight will be presented.

4.1.2 Tracking a reference point

Tracking a reference point at sea involves two steps; the first is to define what reference point to track, the second is to adjust the light to track the specified reference.

4.1.2.1 Reference point calculation

The reference point that is sought to be tracked is defined by the current *POI*. As long as the attitude of the light is not changed by an operator, the *POI* shall be fixed at a certain reference coordinate with respect to the *h-frame*. To express this reference coordinate, transformations had to be done in three steps; between the *h-frame* and the *b-frame*, the *b-frame* and the *l-frame* and finally the *l-frame* end the *POI*. This would yield the *POI* expressed as a reference coordinate with respect to the *h-frame*.

The transformation from the *h-frame* to the *b-frame* depends on the boat's orientation in terms of pitch (β), roll (α) and heave (w) motion. By multiplying the rotational matrices introduced in chapter 3, equation (3.1), and introducing the heave as translation in z -direction the following transformation matrix was given

$$T_h^b(\beta, \alpha, w) = \begin{bmatrix} c(\beta) & s(\beta)s(\alpha) & c(\alpha)s(\beta) & 0 \\ 0 & c(\alpha) & -s(\alpha) & 0 \\ -s(\beta) & c(\beta)s(\alpha) & c(\beta)c(\alpha) & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

When moving between the *b-frame* and the *l-frame* the rotation of the coordinate system will be given by the attitude of the light, defined by its tilt and pan angles, denoted η and γ respectively. Its position is given as a distance from the *b-frame*

origin, denoted as; d_x , d_y and d_z .

$$T_b^l(\eta, \gamma, x_l, y_l, z_l) = \begin{bmatrix} c(\eta)c(\gamma) & -s(\gamma) & c(\gamma)s(\eta) & d_x \\ -c(\eta)s(\gamma) & -c(\gamma) & -s(\eta)s(\gamma) & d_y \\ s(\eta) & 0 & -c(\eta) & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

The complete transformation from the h -frame to the l -frame is then given by multiplying the two transformation matrices in the following order

$$T_h^l = T_h^b T_b^l = \begin{bmatrix} R_{h,xx}^l & R_{h,yx}^l & R_{h,zx}^l & P_{h,x}^l \\ R_{h,xy}^l & R_{h,yy}^l & R_{h,zy}^l & P_{h,y}^l \\ R_{h,xz}^l & R_{h,yz}^l & R_{h,zz}^l & P_{h,z}^l \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Within the transformation matrix T_h^l in equation (4.3) both the position of the light, P_h^l , and its orientation, R_h^l , with respect to the h -frame is given.

The last transformation is to move from the light to the POI . This was done by following the direction of the light's beam until reaching the POI , the direction of the beam being the x -axis of the l -frame. With that said, the following relation was derived

$$\begin{bmatrix} Ref_x \\ Ref_y \\ Ref_z \end{bmatrix} = P_h^l + L_{beam} R_{h,x}^l = \begin{bmatrix} P_{h,x}^l \\ P_{h,y}^l \\ P_{h,z}^l \end{bmatrix} + L_{beam} \begin{bmatrix} R_{h,xx}^l \\ R_{h,xy}^l \\ R_{h,xz}^l \end{bmatrix} \quad (4.4)$$

where L_{beam} is the length of the light beam i.e. the distance between the l -frame origin and the POI .

In order to find L_{beam} , two user cases were defined; one being when the light beam penetrated the surface of the water, the second when it did not. In the first case, the variable Ref_z could be set 0, making it possible to rewrite the equation (4.4) as

$$L_{beam} = -\frac{P_{h,z}^l}{R_{h,xz}^l} \quad (4.5)$$

In the second case, i.e. in situations when the tilt angle was small and the light was pointing towards the horizon, the length of the beam was assumed to be the searchlight's maximum range of 1500 m[1].

Knowing L_{beam} the reference coordinates Ref_x , Ref_y and Ref_z could be found by solving equation (4.4) since P_h^l and R_h^l are known.

4.1.2.2 Light angle calculations

In order to track the calculated reference coordinate, an algorithm was needed to calculate the required pan (γ) and tilt (η) angles. In this section the mathematics behind each angle calculation will be presented together with illustrative figures.

Pan reference angle

To find γ , the reference coordinate was projected onto the xy -plane of the l -frame. The orthogonal projection [13] was achieved by using the following linear algebraic equations

$$Ref_x^{xy} = Ref_x - c_1 \frac{c_1 Ref_x + c_2 Ref_y + c_3 Ref_z + c_4}{c_1^2 + c_2^2 + c_3^2} \quad (4.6)$$

$$Ref_y^{xy} = Ref_y - c_2 \frac{c_1 Ref_x + c_2 Ref_y + c_3 Ref_z + c_4}{c_1^2 + c_2^2 + c_3^2} \quad (4.7)$$

$$Ref_z^{xy} = Ref_z - c_3 \frac{c_1 Ref_x + c_2 Ref_y + c_3 Ref_z + c_4}{c_1^2 + c_2^2 + c_3^2} \quad (4.8)$$

where Ref_x^{xy} , Ref_y^{xy} and Ref_z^{xy} are the projected reference coordinate onto the xy -plane (Ref^{xy}) and c_1 , c_2 , c_3 and c_4 defines the xy -plane as $c_1x + c_2y + c_3z + c_4 = 0$. An example of the projection is illustrated in figure 4.2

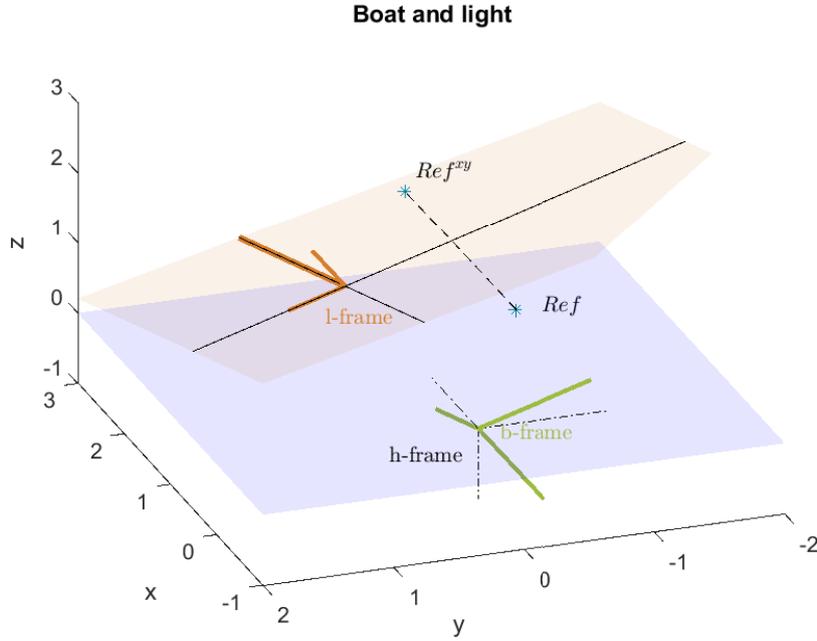


Figure 4.2: Orthogonal projection of reference coordinate onto the xy -plane of the l -frame. The xy -plane is colored orange

The angle γ could then be achieved by calculating the angle between the beam of the light and the projected reference coordinate. Using the definition of the dot product [14], the angle was defined as

$$\gamma = \cos^{-1} \left(\frac{x_l \cdot Ref^{xy}}{\|x_l\| \|Ref^{xy}\|} \right) \quad (4.9)$$

where x_l is the normalized x -axis of the l -frame and also the direction of the light beam.

When using equation (4.9) lack of accuracy will occur at angles close to 0 and π radians, hence equation (4.10) was used instead when implementing the solution in *Matlab*. They are mathematically equivalent but the latter has proven to be more accurate numerically [15].

$$\gamma = \text{atan2}\left(\frac{\|x_l \times Ref^{xy}\|}{x_l \cdot Ref^{xy}}\right) \quad (4.10)$$

Equation (4.10) will give a γ value between 0 and π , since it only calculates the smallest angle between two vectors. By checking the sign of the Ref^{xy} y -coordinate with respect to the l -frame it revealed whether the light should rotate clockwise or counter clockwise, since it indicates which side of the l -frame x -axis the reference coordinate is located at. However, when γ was close to values of π the check could cause the searchlight to rotate almost 2π . This happened in two situations; crossing from just under to just above π and vice versa. To counter this, the algorithm was made aware of its last position, always taking the shortest path to the calculated γ . The resulting rotation created by the angle γ is seen in figure 4.3.

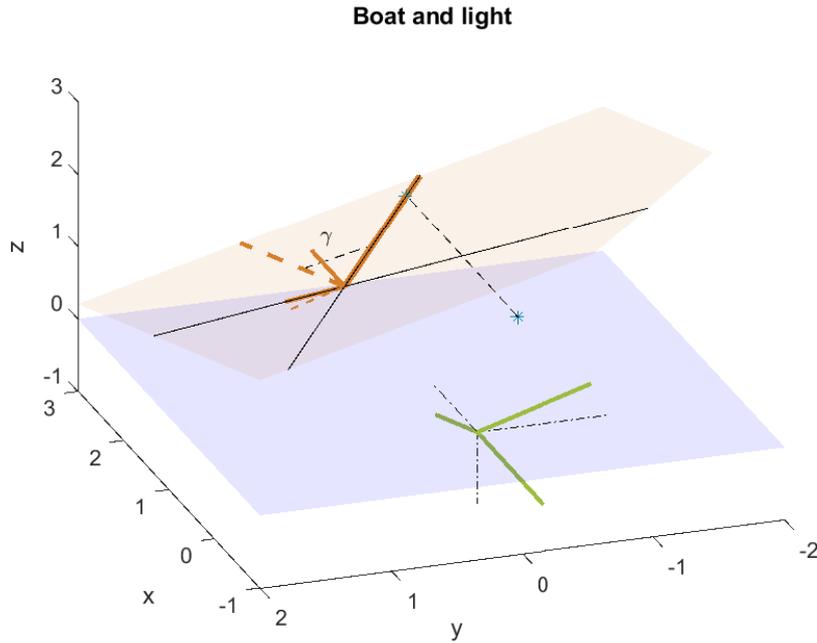


Figure 4.3: Light rotated γ , pointing towards Ref^{xy}

Tilt reference angle

The calculation of tilt angle, η , was done after the rotation of the l -frame with pan angle γ had been performed. The new orientation of the l -frame axes was denoted i_l^{pan} , where $i = \{x, y, z\}$. To calculate the tilt angle the function used in equation (4.10) was used once more as

$$\eta = \text{atan2}\left(\frac{\|x_l^{pan} \times Ref\|}{x_l^{pan} \cdot Ref}\right) \quad (4.11)$$

To define the direction of the angle, the cross product between x_i^{pan} and Ref , seen as the numerator in equation (4.11) was performed. The sign of the y -component of the cross product revealed the correct rotation direction. The resulting rotation is illustrated in figure 4.4.

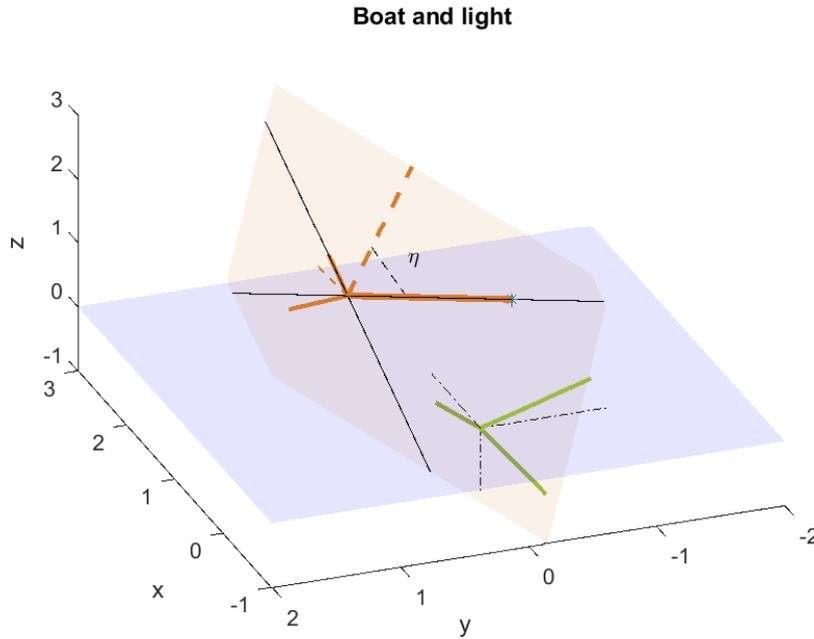


Figure 4.4: Light rotated its calculated tilt angle η

4.2 Control system design

From the equations representing the dynamics of the system found in section 3.3.1, a state space representation could be determined. In this section the state space as well as an analysis will be presented followed by the design of the controllers.

4.2.1 State space representation

In order to control a system using LQR and LQI controllers, the system must be expressed on a state space form. Three states were chosen to represent the system, the motor current, the motor angle and the motor angle velocity, as follows

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} i \\ \theta \\ \omega \end{bmatrix} \quad (4.12)$$

The current, i , was chosen since it appears as its derivative in equation (3.3), which also was the case for the angle θ . However, the angle had the additional condition that it constitutes the output of the system, hence it had to be included in the state vector. The voltage, v , was chosen as input, u , to the system. By introducing the

states, three first order equation could be derived, based on the equations presented in section 3.3.1 in chapter 3, as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{i} \\ \omega \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{v}{L} - \frac{Rx_1 - K_u}{L} - \frac{K_u x_3}{L} \\ x_3 \\ \frac{K_m x_1}{J} - \frac{bx_3}{J} \end{bmatrix} \quad (4.13)$$

The equations in (4.13) are linear time invariant (*LTI*) and could consequently be represented directly in state space form with matrices as follows

$$A = \begin{bmatrix} \frac{-R}{L} & 0 & \frac{-K_u}{L} \\ 0 & 0 & 1 \\ \frac{K_m}{J} & 0 & \frac{-b}{J} \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \end{bmatrix} \quad C = [0 \quad 1 \quad 0] \quad D = 0 \quad (4.14)$$

To get an understanding of the system, the state space representation was analyzed to check stability and reachability. The actual parameters of *CLITE 2* was not obtained so an estimation of the data was done and seen in table 4.1. The assumptions and calculations made to produce the parameters is described in A.4.

Light Parameters			
Parameter	Variable name	Unit	Value
Resistance	R	Ω	0.6
Inductance	L	mH	2.1
Motor constant	K_m	$\frac{Nm}{\sqrt{W}}$	0.45
Back-EMF constant	K_u	Vs	0.35
Inertia	J	kgm^2	0.01
Damping coefficient	b	$\frac{Nm}{KRPM}$	0.01

Table 4.1: Light parameters used to verify system

The above parameters were introduced in the matrices in equation (4.14) to form the following state space matrices.

$$A = \begin{bmatrix} -300 & 0 & -175 \\ 0 & 0 & 1 \\ 45 & 0 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 500 \\ 0 \\ 0 \end{bmatrix} \quad C = [0 \quad 1 \quad 0] \quad D = 0 \quad (4.15)$$

The motor equations (3.3) and (3.5) presented in chapter 3 were expressed in continuous time, why the state space matrices were also continuous. The system had to be converted in to a discrete time system, in order to be implemented on a searchlight. This was done by using the discrete time state space representation as

$$\begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k) \end{aligned} \quad (4.16)$$

In order to achieve the discrete time matrices A_d , B_d , C_d and D_d the following transformation was applied

$$\begin{aligned}
 A_d &= e^{AT_s} = \begin{bmatrix} -0.0149 & 0 & -0.4893 \\ 0.001 & 1 & 0.0093 \\ 0.1258 & 0 & 0.821 \end{bmatrix} \\
 B_d &= \int_0^{T_s} e^{A\tau} B d\tau = \begin{bmatrix} 1.4087 \\ 0.0020 \\ 0.4849 \end{bmatrix} \\
 C_d &= C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\
 D_d &= D = 0
 \end{aligned} \tag{4.17}$$

The sampling time, T_s , was set to 0.01 s . This was chosen since it was proven to be the fastest manageable sampling time for the processors used later in the report. Since the matrix A^{-1} does not exist for this particular system, the commonly used method for approximating the integral by $A^{-1}(A_d - I)B$ cannot be used, instead numerical integration can solve the problem.

The transfer function from the input to the output was defined as

$$\begin{aligned}
 G(z) &= C_d(zI - A_d)^{-1}B_d + D_d \\
 &= \frac{1.4 \cdot 10^{-6}z^2 + 2.2 \cdot 10^{-6}z}{z^3 - 2z^2 + 1.001z - 0.0006}
 \end{aligned} \tag{4.18}$$

To check whether the system was stable the poles of the transfer function was calculated by setting the denominator to 0 and solve for z

$$z^3 - 2z^2 + 1.001z - 0.0006 = 0 \tag{4.19}$$

which yielded

$$poles = \begin{bmatrix} 1 \\ 0.0667 \\ 0.7394 \end{bmatrix} \tag{4.20}$$

In order to be stable, the entire set of poles have to be strictly within the unit circle. In this case, one pole was situated on the circle, meaning that the open loop system was Lyapunov stable but not asymptotically stable [11].

4.2.2 Controllers

In order to achieve the desired behaviour of the system, i.e. to satisfyingly track a reference coordinate, the controllers had to be implemented and tuned. With the help of simulations, presented in section 4.3, the three controllers' parameters were set. These parameters will be presented in this section together with a brief stability analysis of the system.

4.2.2.1 PID

The parameters P , I and D were tuned by trial until satisfying performance was achieved. The resulting values were

$$\begin{aligned} P &= 10 \\ I &= 0.1 \\ D &= 0.0 \end{aligned}$$

As seen above, the value of D was set to 0 since the impact of the derivative part did not contribute to the performance of the closed loop system. The zeros and poles of the closed loop system are seen in figure 4.5.

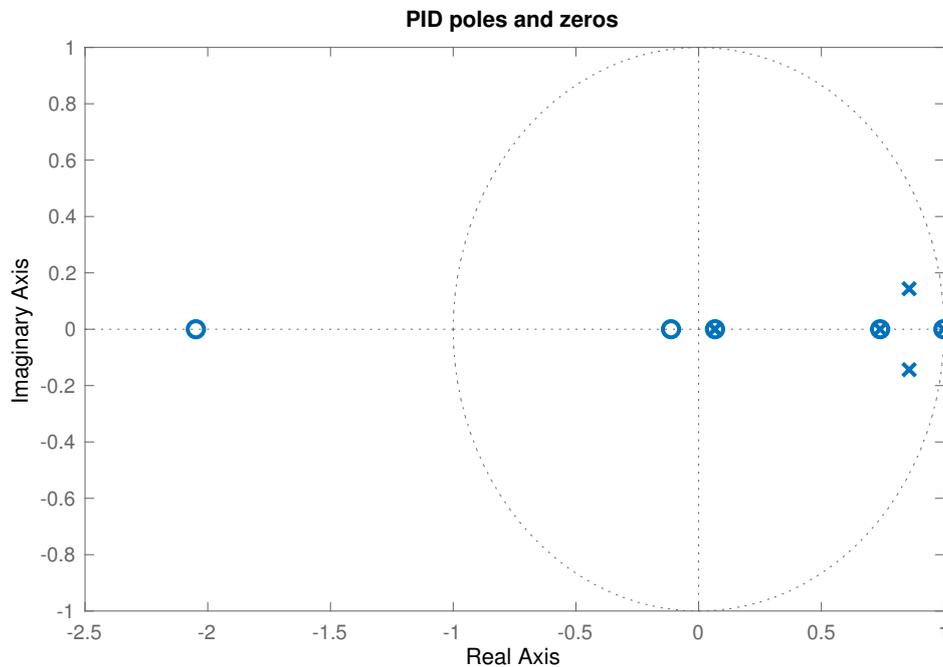


Figure 4.5: Zeros (o) and poles (x) of the closed loop system using PID controller.

4.2.2.2 LQR

As described in section 3.3.3, the system must be reachable to achieve an optimal feedback gain. With the values from table 4.1 in section 4.2.1 above, the reachability matrix was evaluated as

$$R_m = \begin{bmatrix} B_d & A_d B_d & A_d^2 B_d \end{bmatrix} = \begin{bmatrix} 1.4087 & -0.2583 & -0.2776 \\ 0.0020 & 0.0078 & 0.0129 \\ 0.4849 & 0.5753 & 0.4399 \end{bmatrix} \quad (4.21)$$

The matrix R_m is full rank, which means that an optimal feedback gain could be obtained for this particular system. The design parameters chosen for the matrices

4. Method

Q_u and Q_x are shown below

$$Q_x = \begin{bmatrix} 1 * 10^{-2} & 0 & 0 \\ 0 & 1 * 10^{-6} & 0 \\ 0 & 0 & 5 * 10^{-4} \end{bmatrix} \quad (4.22)$$

$$Q_u = [1 * 10^{-5}]$$

As can be seen in equation (4.22), the design parameters are tuned so that the current is the penalized the most out of the three states. Although the other two are relatively small they still effected the output heavily. Regarding the penalizing of the output it is also small, resulting in a control strategy being somewhat expensive since Q_u is bigger than $x^T Q_x x$.

With the design parameters chosen, the resulting feedback gain K was calculated using equation (3.12) as

$$K = [0.0075 \quad 0.0068 \quad -0.2105] \quad (4.23)$$

With the given K -value, the pole placement for the closed loop system is seen in figure 4.6

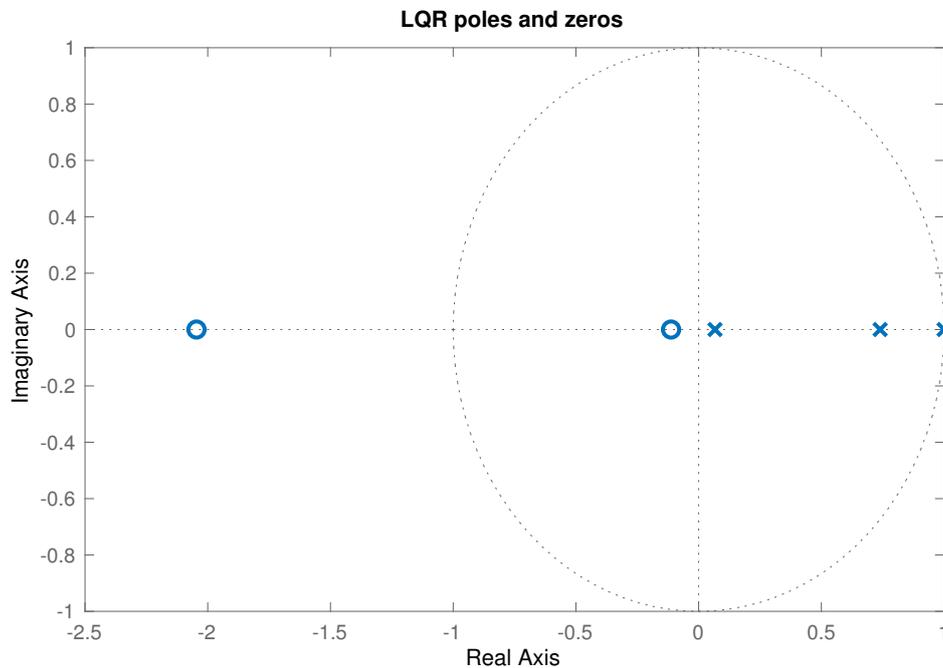


Figure 4.6: Zeros (o) and poles (x) of the closed loop system using LQR controller.

where all poles were strictly within the unit circle. Hence the system was both Lyapunov- and asymptotically stable.

4.2.2.3 LQI

With R_m being full rank, the *LQI* controller could be implemented on the control the system. By tuning the matrices Q_u and Q_x as seen in equation (4.24) the feedback gain K_i shown in (4.25) was achieved.

$$Q_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 * 10^{-6} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix} \quad (4.24)$$

$$Q_u = [0.001]$$

In this case the $x^T Q_x x$ was significantly bigger than Q_u , indicating a cheap control strategy. The calculated feedback gain K_i was calculated using the augmented matrices in equation (3.12) seen below

$$K_i = [0.061 \quad 17.304 \quad 0.424 \quad -189.797] \quad (4.25)$$

Calculating the pole placement for the closed loop system using K_i as feedback gain yielded the result seen in figure 4.7

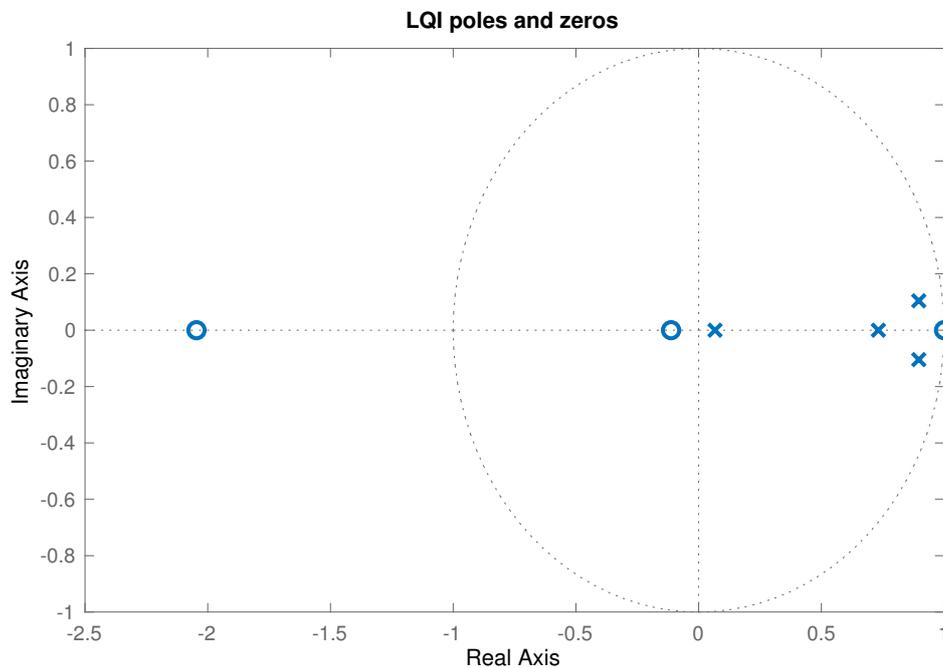


Figure 4.7: Zeros (o) and poles (x) of the closed loop system using *LQI* controller.

where all poles are strictly within the unit circle proving the system to be both Lyapunov- and asymptotically stable.

4.3 Simulations

After the algorithms and the control solutions had been created, simulations were made in the graphical programming environment *Simulink*. The attitude of the searchlight was emulated by feeding the system with a pan and tilt angle. The system was then fed with arbitrary boat motions in the form of pitch, roll and heave. After receiving the external data, the input to the system was calculated into a Cartesian reference coordinate with help of the previously defined mathematical model in section 4.1.2.1. This coordinate, together with the orientation of the boat, got calculated into reference angles for the lamp to position itself to, as defined in section 4.1.2.2. The desired angles then got fed into the different controllers. The output signals were then used as input to a transfer function, in the case of the *PID*, and a state space representation for the *LQ* controllers. These aimed to emulate the two electric motors and the systems inherent dynamics, resulting in an output of the angles γ and η .

A stripped down version of the model used for simulations is seen in Figure 4.8. For a more detailed view, see Appedix A.1 and A.2.

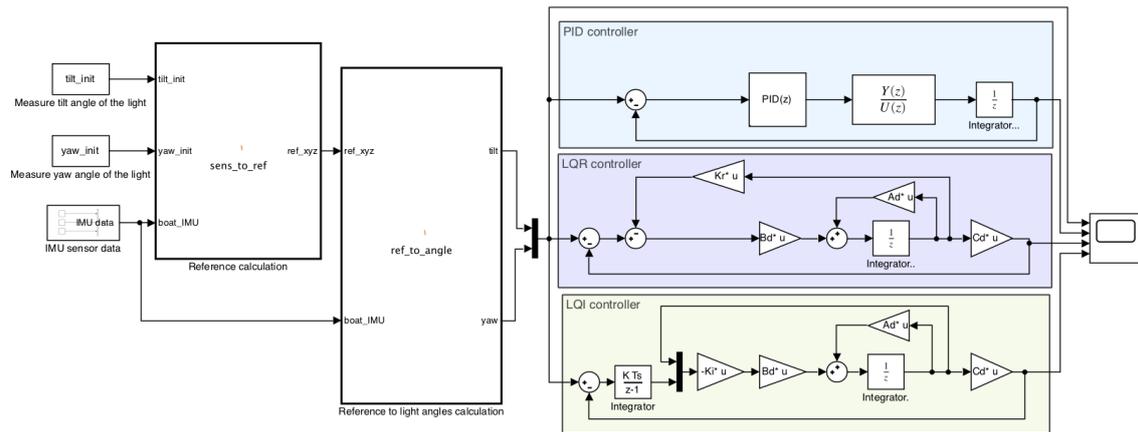


Figure 4.8: Simplified overview of the simulation setup as implemented in *Simulink*.

4.4 Physical testing

When the mathematical model and the three controllers functionality had been verified with simulations, the project proceeded to implement the same functionality on a physical system.

4.4.1 Test equipment

In order to test the intended functionality, a simple version of a searchlight was constructed with the help of two servo motors, a laser pointer, two potentiometers and an *IMU*. The mounted test system can be seen in figure 4.9, where the potentiometers are found inside the two servo motors, acting as the *CLITE 2*'s rotary

encoders. A list describing all components is found in Appendix A.3. The two servo motors together with the potentiometer made it possible to control pan and tilt of the light. The *IMU* measured the movement of the simulated boat.

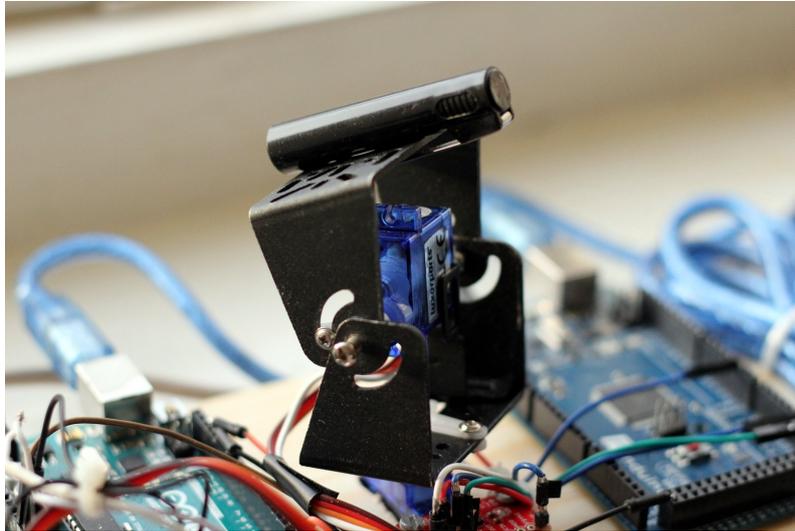


Figure 4.9: Test rig used to emulate the two engines in the *CLITE 2*

The data provided from the *IMU* sensor was provided to the algorithm as a relatively smooth signal, with the help of *Kalman* filtering [16] and a manageable raw signal from the sensor. The signal from the potentiometers was also filtered with the help of a *Kalman* filter.

The system was setup by letting *Simulink* run code live on a *Raspberry Pi 3 model B*. The positions of the servos were then read by feeding back the potentiometers' values. The *Raspberry Pi* was not able to read analog signals; so the potentiometer readings were fed to an *Arduino UNO* in order to take advantage of its analog to digital converter. These converted values were then sent via serial port to the *Raspberry Pi*. An overview of the system can be seen in figure 4.10.

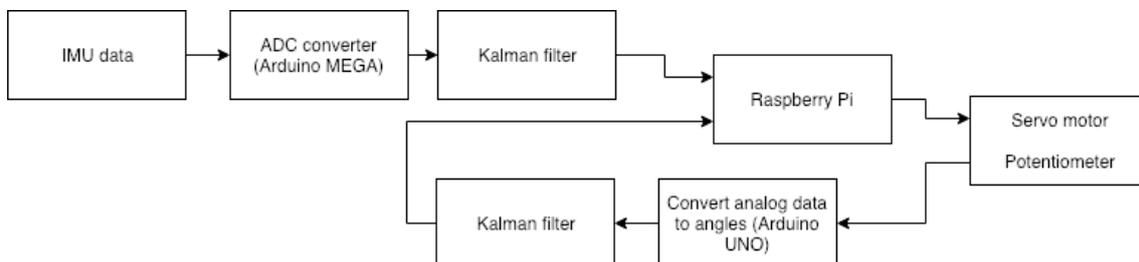


Figure 4.10: Overview of the implemented hardware

The calculated control parameters, for the three controllers, in section 4.2.2 aimed to control the characteristics of the *CLITE 2*. However, since the physical tests were performed on a different set of hardware, these values had to be altered. The new values were achieved by analyzing the behaviour of the system while exposed to wave motions and modified accordingly.

4.4.2 Test procedure

The testing was done by placing the test equipment on a 6 degrees of freedom ($6DOF$) platform, as seen in figure 4.11. The platform was able to emulate yaw, pitch, roll and translation in x , y and z directions. Usually being used to emulate the dynamics of a car, it was instead fed with sinusoidal waves for pitch and roll, in order to mimic the motion of the sea. These sinusoidal motions interpreted by the the rig's IMU is to be seen in figure 4.12.



Figure 4.11: The test rig placed on a $6DOF$ platform for testing

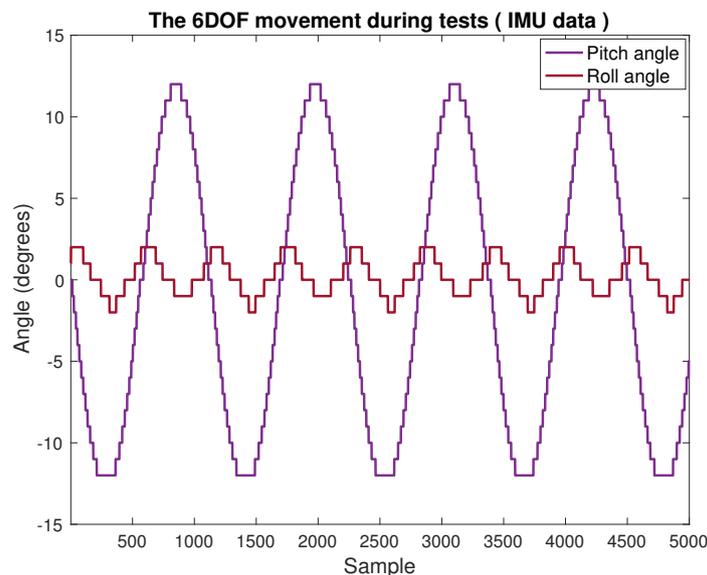


Figure 4.12: The movement of the $6DOF$ platform when testing the controllers.

While moving, the calculated reference angle and the measured motor angle for both motors were stored. The POI was also captured by using a camera, mounted on a tripod, and photographing a sequence of the test with a 20 second shutter speed. In

order to give a sense of distance, the area hit by the light beam were marked with a vertical and a horizontal scale, both marked with lines with 5 *cm* spacing, as seen to the left in figure 4.11. The setup is seen sketched in figure 4.13 below.

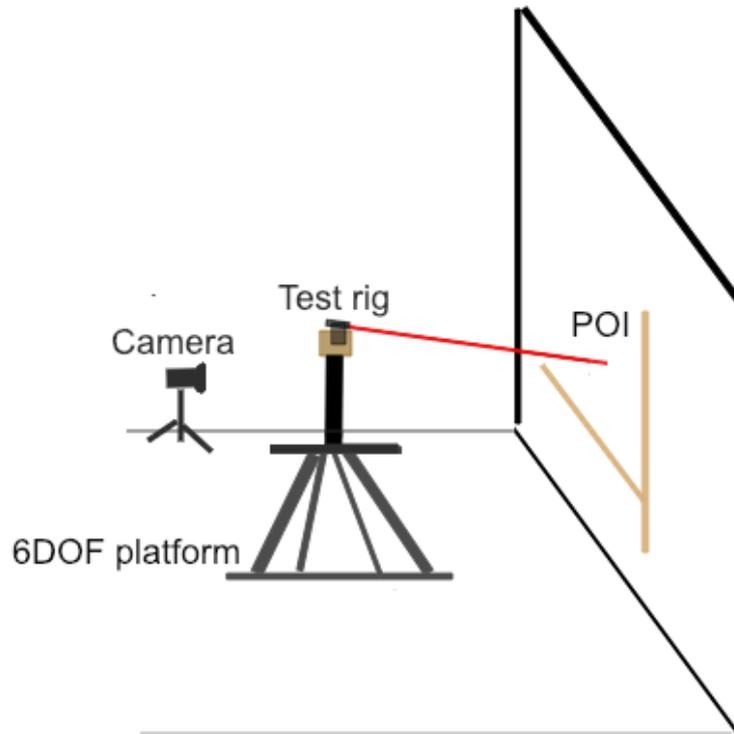


Figure 4.13: Sketch of the test setup used during physical testing.

5

Results

In this chapter the results of the project will be presented in the form of simulation results and results of testing on a physical system. The plots illustrate how well the control system tracks a reference angle in order to keep the *POI* at the calculated reference coordinate. The results from the physical tests are also presented in photos, showing the *POI* over time.

5.1 Simulation

In figure 5.1 a reference coordinate, that is changed after five seconds, can be seen. This acted as an input to the system which the simulated result were based upon. To its right, in figure 5.2, the boats movements in the form of pitch, roll and heave are plotted together with the resulting light angles, γ and η .

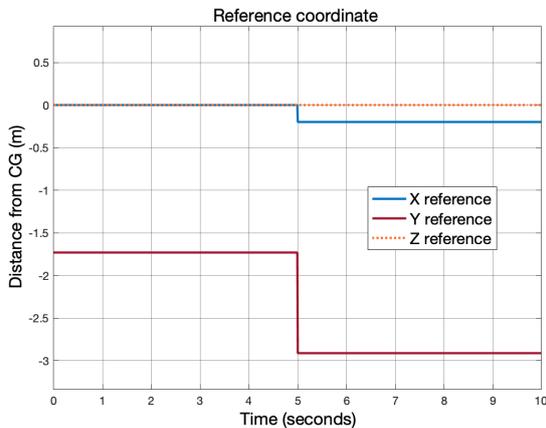


Figure 5.1: Reference which the searchlight is set to point at with a new reference being set after 5 seconds.

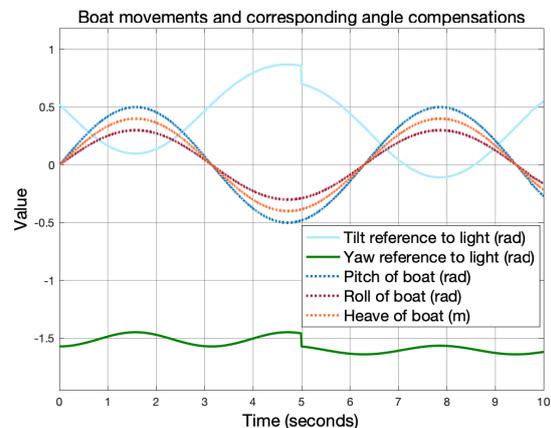


Figure 5.2: Boat motion and the resulting light angles calculated by the system based on the reference coordinate in figure 5.1.

Three different controllers were implemented and tuned. In figure 5.3 the step response of these three controllers can be seen together with a close up of their behaviour in 5.4. In figure 5.5 the different controllers ability to track the reference angles seen previously in figure 5.2. In figures 5.6 and 5.7 the squared error between the calculated reference angles and the controller output is presented.

5. Results

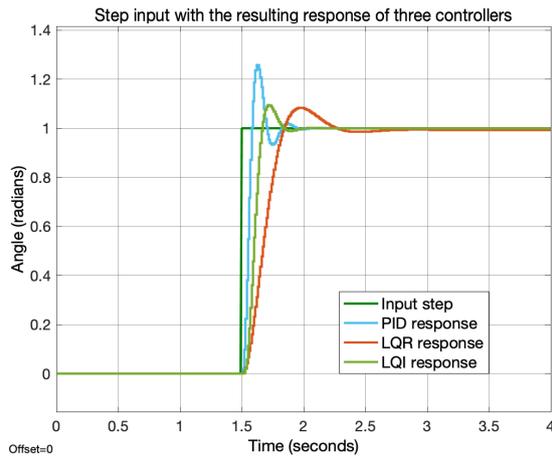


Figure 5.3: The three controllers response to a step input.

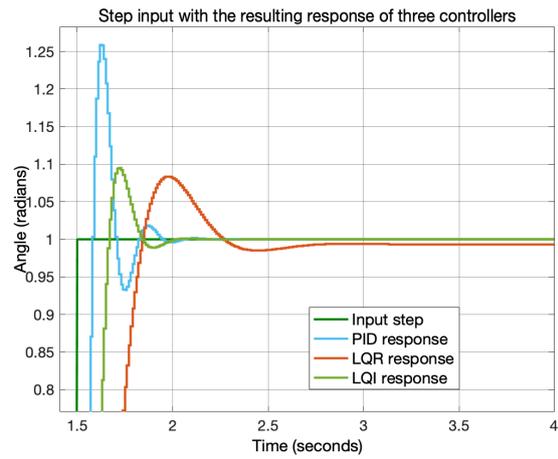


Figure 5.4: A close up of the three controllers response to a step input.

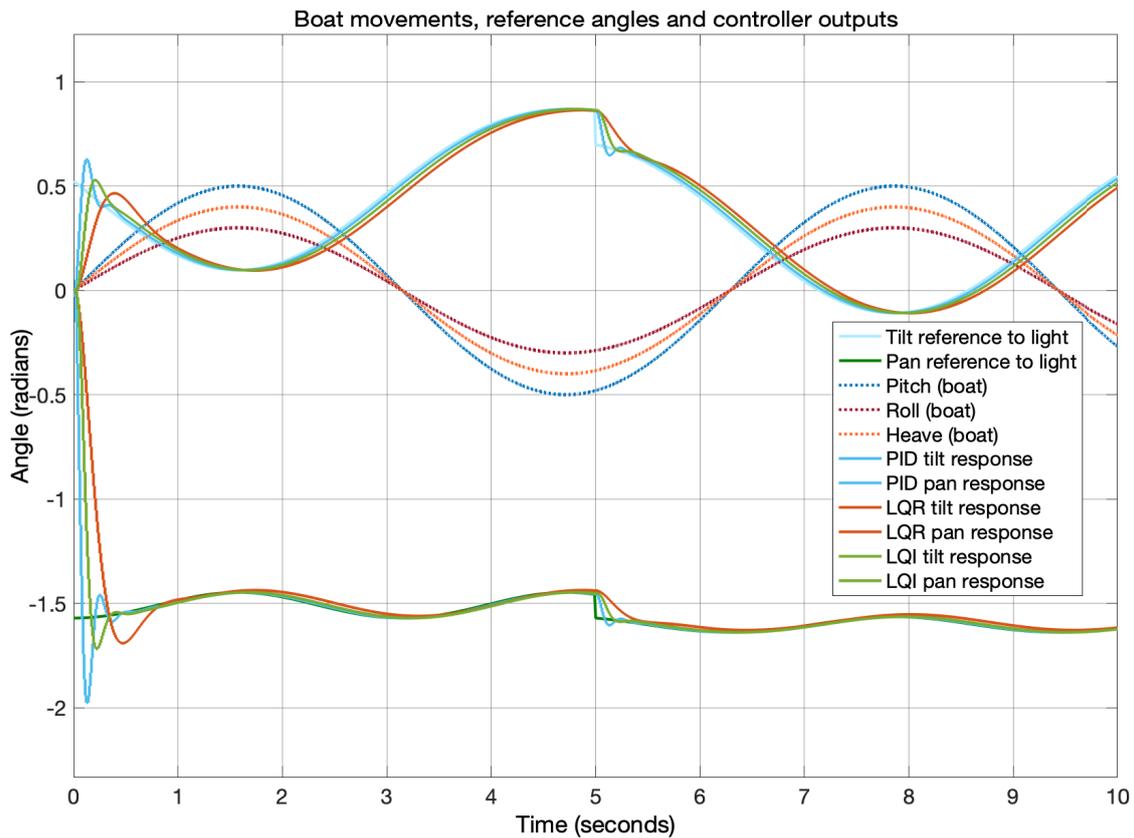


Figure 5.5: The controllers ability to track moving reference angles as a result of the boats motion.

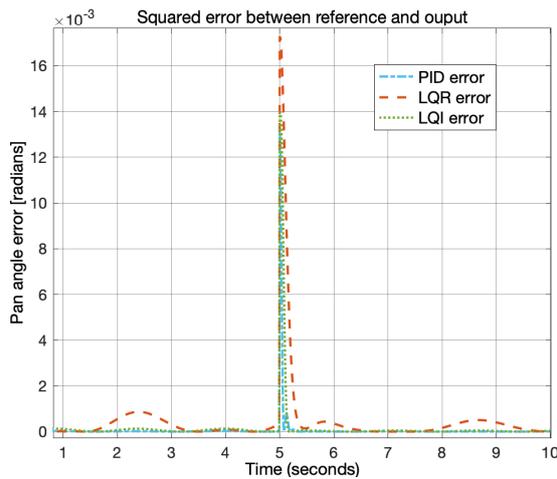


Figure 5.6: The squared error between the pan angle reference and controller output.

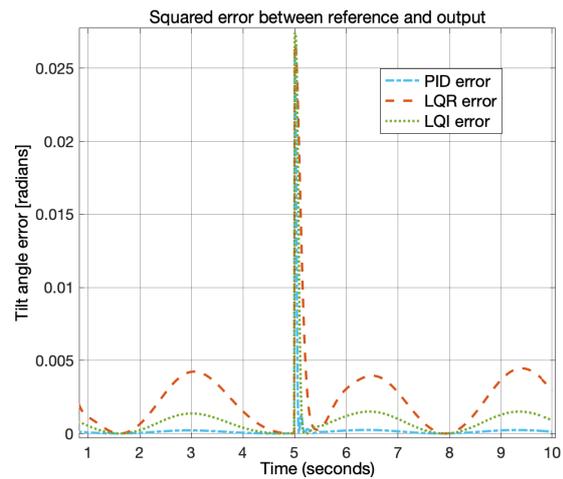


Figure 5.7: The squared error between the tilt angle reference and controller output.

The data presented in figures 5.6 and 5.7 above was also recalculated in the form of *minimum mean square error* (MSE). The resulting values is seen in table 5.1 below.

MSE values for control of pan and tilt (10^{-4})			
	PID	LQR	LQI
Pan	0.71	4.82	1.64
Tilt	2.40	24.05	9.46

Table 5.1: MSE values for the error between the reference and the controller output.

5.2 Physical testing

When the algorithm and controller's functionality had been verified in simulations, the system was tested on hardware with the help of a $6DOF$ platform. In figures 5.9 and 5.8 below the response of the pan and tilt motors for the three different controllers are presented together with the reference angles. These plots are the results of measuring the position of the motors and logging the reference angles calculated by the algorithm. In table 5.2 the MSE values for the error between the reference angle and control output angle is presented.

5. Results

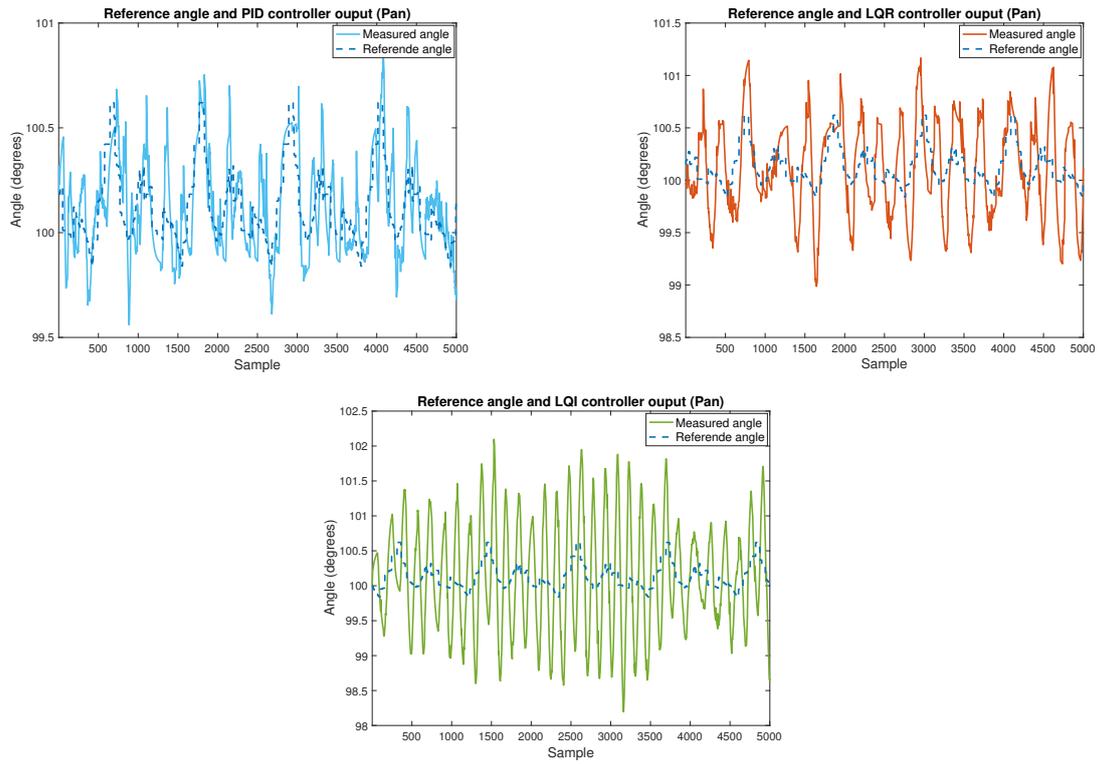


Figure 5.8: The reference pan angle compared with the measured angle for three different controllers during physical testing.

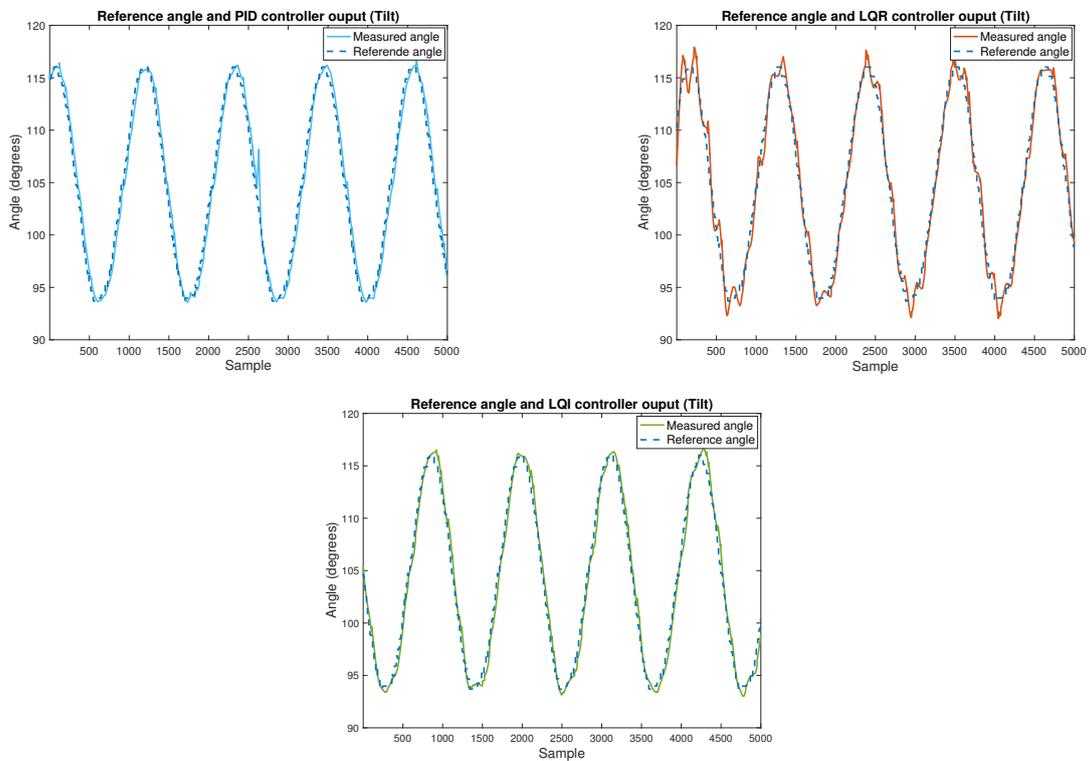


Figure 5.9: The reference tilt angle compared with the measured angle for three different controllers during physical testing.

MSE values for control of pan and tilt (10^{-1})			
	PID	LQR	LQI
Pan	0.29	1.60	6.36
Tilt	10.90	13.09	7.55

Table 5.2: MSE values for the error between the reference and the controller output.

To get a better understanding of the effect of the different control systems the *POI* was photographed and presented in figures 5.10 - 5.13.

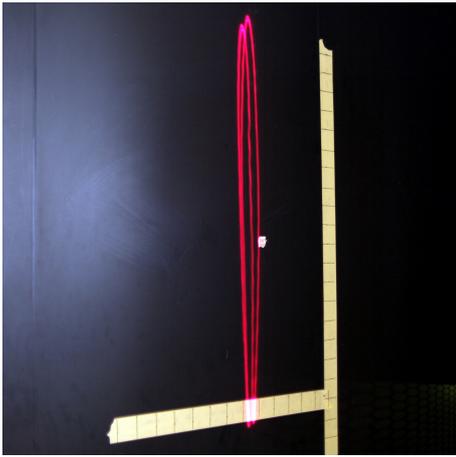


Figure 5.10: Light *POI* for the system without any controller.

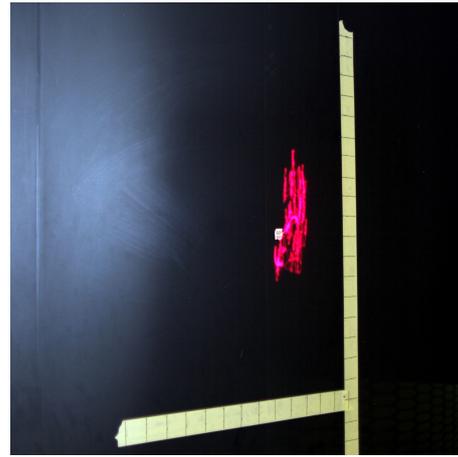


Figure 5.11: Light *POI* for the system with PID controller.

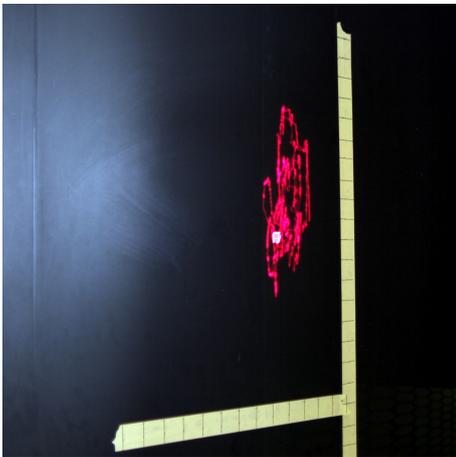


Figure 5.12: Light *POI* for the system with *LQR* controller.

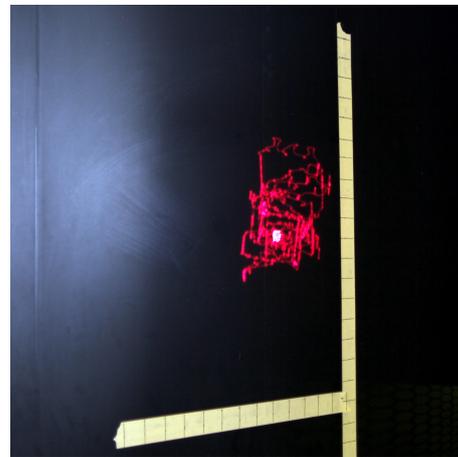


Figure 5.13: Light *POI* for the system with *LQI* controller.

6

Discussion

6.1 Mathematical model

When calculating the length of the light beam in equation (4.4) the z -coordinate is either set to 0, meaning that the POI is at the water surface, or does not hit the water within its range of 1500 m . However, there are circumstances when none of these situations are true. For example when a user wants to light up a ship in the water which is leveled on the same height as the position of the searchlight. The algorithm will perceive this situation by stabilizing the POI at a distance of 1500 m while the ship might be situated closer. This would lead to the POI not being perceived as stable. This kind of situations could likely be solved by introducing sensor able to measure distance, e.g. a sonar sensor or radar. The sensor could then reveal the parameter L_{beam} in equation (4.4) which in turn would yield the correct reference coordinates.

Regarding the trigonometry of the light, it does not account for a distance between the motor and the actual light source. If there would be a distance between them, a change in angle sent to motor would lead to a shift in the light source position. I.e. the l -frame origin would move, adding a heave like effect to the light. Depending on the design of a searchlight, this might affect the resulting behaviour of the system. Though, regarding the specific searchlight *CLITE 2*, this was not considered a problem.

The control system takes the movements in roll, pitch and heave as inputs, while surge, sway and yaw were not accounted for. The reason was it not being deemed necessary in order to stabilize the light for the chosen method. A compensation for yaw would also mean that a separation between a yaw caused by the act of turning the boat and a yaw induced by waves would be needed. To manage that, information from the rudder angle would be beneficial in order differentiate the two. The same reasoning applies for surge and sway.

6.2 Simulation results

The performance of the system seen in the simulation plots in section 5.1, was mainly dependent on the tuning of the controllers. In figure 5.3 and 5.4 the step response of the three different controllers are presented. From the results one can note that the PID had the shortest rise time, but also experienced the largest overshoot. The LQI

had a similar rise time but a significantly smaller overshoot, followed by the *LQR* with a relatively slow rise time, but with an overshoot comparable with the *LQI*. The overshoot, regarding all controllers, was however a result of their tuning. It was required from the system to be fast enough to track the constantly changing reference angle, which made it difficult to avoid this kind of behaviour. One might also note that a considerable amount of time was spent tuning the three controllers, and the results presented represents the best performance that was able to be achieved in the window of time given. In the figure 5.4, showing the close up of the step response, it can be seen that the *LQR* does not only show the worst performance in rise time, but it also never reaches the reference value of 1. This is most likely due to that the *LQR* controller does not contain an integrator, which would handle the constant error. Worth mentioning is that tuning of the *LQR* and *LQI* controllers were considerably more time consuming than the tuning of the *PID*. The *PID* gave satisfactory results within a couple of minutes, while the others had to be tuned for hours to even be compared to the *PID*.

In figure 5.6 and 5.7, the squared error between the reference and the controller output was presented. In the figures it becomes evident that the best performing controller, during simulations, was the *PID*, followed by the *LQI* and then the *LQR*. This also became even more clear when analyzing the *MSE* values of the three controllers where the *PID* showed superior performance compared to the other controllers. This result could mean that the *PID* is a better performing controller for the task at hand, but it could also be a result of better tuning. As mentioned, the tuning of the *LQR* and *LQI* controller was a tedious task. A further perfection of the controller's parameters could possibly have led to better results than the one presented in this thesis.

6.3 Physical testing

When testing the mathematical model together with the controllers on a physical system the results became heavily dependent on the functionality of the hardware and the tuning of the system. Nonetheless, from the results presented in figures 5.10-5.13, showing the *POI* with and without controllers, it is clear that the system is proving to track a reference point, albeit not perfectly. Although the simulations also proved that a perfectly tracking system probably is not possible, the physical results were believed to be able to perform even better, and the reasons for this will be presented below.

6.3.1 Hardware

The single largest difference between the simulations and the physical tests was the actual hardware, since it was not the same as the modelled system. Mainly because the control system was not able to be implemented on the *CLITE 2* and since a model of a real system rarely is an exact description of its physical equivalent. As mentioned in chapter 4, the feedback signals from the motors and the *IMU* data were noisy, and filtering of the signals was necessary. Attempts of reducing the noise

were made with *RC*-filter circuits and *FIR*-filters, but the results were not satisfying. In the end a Kalman filter was implemented due to its compromise between smoothing and delay of the signal. The delay, however, made the system less responsive to sudden changes. Since all controllers were tested with the same delay this did not effect the comparison between the three. However, the delay most likely affected how well the system performed in tracking a reference point when subjected to disturbances.

The overall hardware setup could be seen as contributing to an under performing system. With the system being tested on primitive hardware, improvements such as using more precise rotary encoders instead of feeding back signals from the motor potentiometers, would reduce the uncertainty and need of the same level of filtering. The motors also suffered from jitter resulting in the system not being able to track a point perfectly even when the system not being subjected to any movements.

6.3.2 Tuning of controllers

The tuning of the system was vital to the performance. The *PID* was relatively easy to tune for the physical system. The *LQR* and *LQI* controllers on the other was harder to dial in. This increased the time for tuning considerably. In the end, all controllers were more or less tuned to behave similarly when subjected to a sinusoidal wave, but there is no guarantee that the tuning made actually was the best there could be.

Analyzing the result of the tilt angle tracking in figure 5.9 it becomes evident that all three controllers were able to track the calculated reference fairly well, with the *LQR* controller being less smooth than the other two. With the pan angle tracking seen in figure 5.8, the angle differences are much smaller, and frequency higher. Despite that, the *PID* is seen tracking the reference angle rather well, followed by the *LQR* and then the *LQI* which tracks the reference angle fairly poor. Since the *LQI* is seen tracking the tilt reference angle well, this is possibly a result of poor tuning, or hardware related as discussed earlier. It was nonetheless a sign of the controller's inability to track a fast moving reference angle. This was also seen, for all controllers, when the frequency of the sinusoidal wave being fed to the *6DOF* platform was increased. Doing so, all controllers performed poorly to the extent that they did not visibly stabilize the searchlight.

The reason why the *LQI* was performing worse than the other controllers regarding pan angle tracking could be explained by the added integral state, which requires a number of samples in order to accumulate an error. For this reason the tracking of a slowly changing reference could perform less preferable. On the other hand, thanks to the integral state, a constant tracking error would be eliminated. Possibly the design parameters, Q_u and Q_x , should have been tuned in a better way such that the benefits of the error elimination overcomes the downsides of its poor tracking of small changes.

6.3.3 Testmethod

The *6DOF* platform was only programmed to pitch and roll, not to heave even though the algorithm was designed to handle pure heave motions. Although this was verified during simulations, the hardware did not support measurement of heave. Attempts were made to use the *IMU* to estimate the heave displacement, which involved integrating the accelerometer data twice. Unfortunately the accelerometer signal proved to be too noisy and as a result the distance in heave that was estimated was not useful. To be able to estimate heave another sensor would most likely be needed in the system and possibly fused with the information from the accelerometer. It has however been done by only using an *IMU*, as seen in the report *Heave Motion Estimation of a Vessel Using Acceleration Measurements* [17].

The platform was intended to mimic a boat at sea during the tests. It was done by introducing sine waves in the *6DOF* platform software for both roll and pitch, since wave motion in general is often simulated as sinusoidal functions [18] [19] [20] [21]. Furthermore, the transfer function between the wave motion and the motion of a boat is sometimes modeled as a state space filter [22], similarly the signal between the sinusoidal input and the motion of the *6DOF* platform was filtered. Despite the fact that the platform was programmed for other purposes than ship modeling, one can equate the physical testing with a boat at sea to some extent.

6.4 Implementation on the *CLITE 2*

In the case of implementing the control system on the *CLITE 2*, better results than the ones seen in this thesis is expected. This is mainly due to the lights better performing motors and use of encoders, with presumably better accuracy than the use of potentiometers. An implementation would however entail introducing an *IMU* to the light and making sure that its processor can handle the needed calculations.

If testing would be done on an actual boat in future the control system will most likely stabilize the searchlight. However, how the user would experience the stabilization of the light is unknown. The anticipation is that the algorithm, on a well functioning system, would help facilitate the use of the light, but there is no guarantee that it would be the case. The control system could possibly be perceived as obstructing, rather than assisting.

7

Conclusion

This master's thesis was based on the objective of developing a control system for stabilizing a searchlight, a task with many possible solutions and outcomes. After long discussions, both externally and internally a choice to track a reference point, relative the boat was made. Based on this idea the mathematical model was built and later implemented. Although not perfect, the algorithm together with the controllers proved that a searchlight could be stabilized based on *IMU* data and with feeding back the positions of the motors. With that said, further testing needs to be done in order to ensure that the proposed control system satisfies the needs of a possible end user.

Bibliography

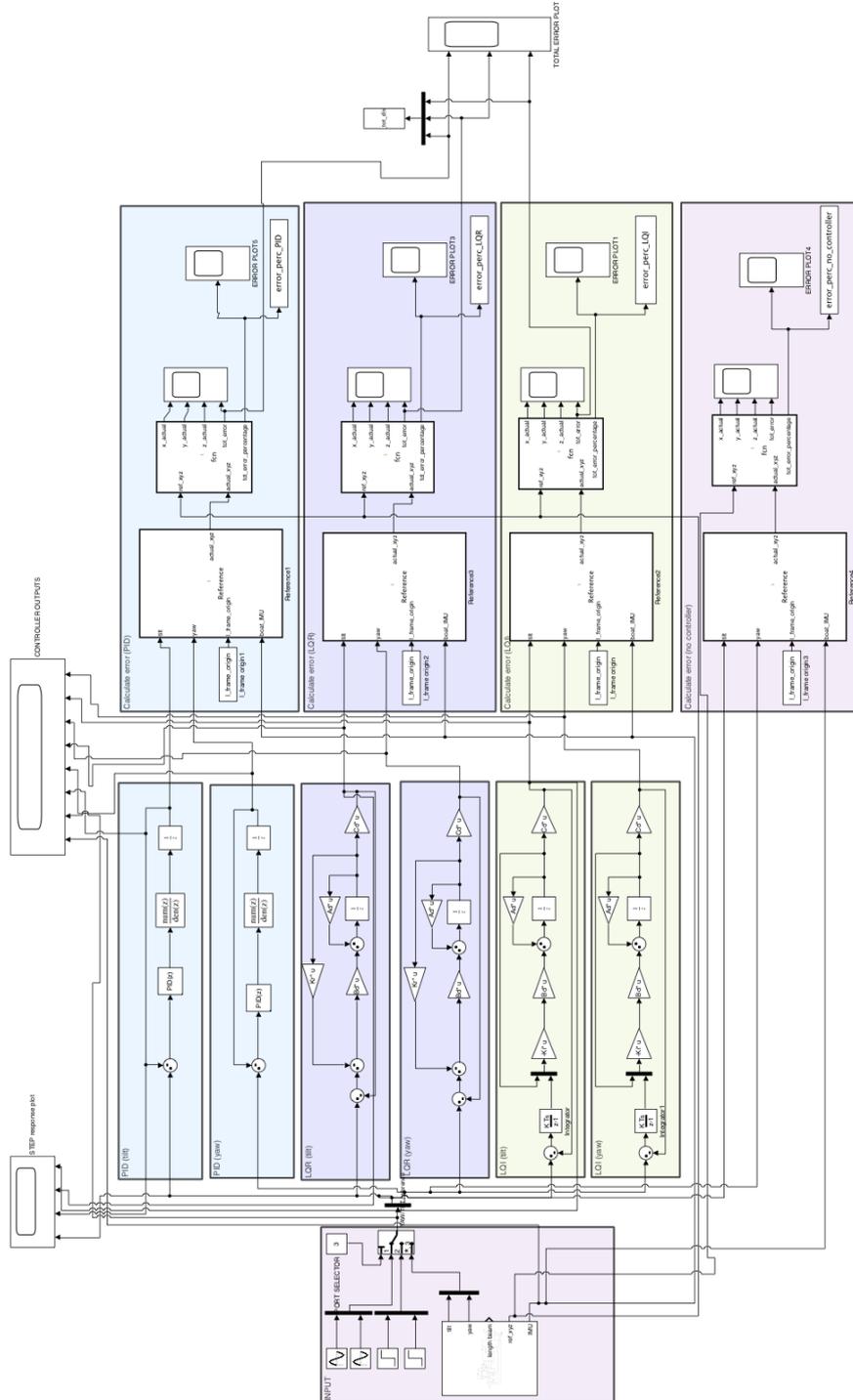
- [1] Luminell, “Clite led searchlight.” [Website]. Available: <https://luminell.com/products/searchlights/clite2-searchlight/>. Retrieved 07-Sep-2018.
- [2] R. Antonello, R. Oboe, D. Pilastro, S. Viola, K. Ito, and A. Cenedese, “Imu-based image stabilization in a hsm-driven camera positioning unit,” in *2013 IEEE International Conference on Mechatronics (ICM)*, pp. 156–161, Feb 2013.
- [3] G. Kumar, P. Y. Tiwari, V. Marcopoli, and M. V. Kothare, “A study of a gun-turret assembly in an armored tank using model predictive control,” in *2009 American Control Conference*, pp. 4848–4853, June 2009.
- [4] L. Guo and Z. Tan, “Application of a sliding mode variable structure controller for stabilized platform of shipborne weapons,” in *2010 2nd IEEE International Conference on Information Management and Engineering*, pp. 65–68, April 2010.
- [5] Y. Xia, L. Dai, M. Fu, C. Li, and C. Wang, “Application of active disturbance rejection control in tank gun control system,” *Journal of the Franklin Institute*, vol. 351, no. 4, pp. 2299 – 2314, 2014. Special Issue on 2010-2012 Advances in Variable Structure Systems and Sliding Mode Algorithms.
- [6] “Gimbal.” [Encyclopedia]. Available: <https://www.encyclopedia.com/social-sciences-and-law/political-science-and-government/military-affairs-nonnaval/gimbal>. Retrieved 03-01-2019.
- [7] A. Boström, *Rigid body dynamics*. Chalmers, Gothenburg: Anders Boström, October 2013.
- [8] D. Bell, *Fundamentals of Electric Circuits*. Oxford University Press, 2009.
- [9] J. Matolyak and A. Haija, *Essential Physics*, pp. 65–84. CRC Press, 2013.
- [10] K. Aström and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*, pp. 216–251. Princeton University Press, 2010.
- [11] R. Murray, Z. Li, S. Sastry, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, pp. 43–53. Taylor & Francis, 1994.
- [12] K. J. Åström and R. M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton, NJ, USA: Princeton University Press, 2008.
- [13] R. Adams and C. Essex, *Calculus: A Complete Course*. Pearson Canada, 2009.
- [14] S. Lipschutz and M. Lipson, *Schaum’s Outline of Linear Algebra, Sixth Edition*. McGraw-Hill Education, 2017.
- [15] *Angle between two vectors*. 2007. [Forum Discussion], Available: <https://groups.google.com/d/msg/comp.soft-sys.matlab/zNbUui3bjcA/zfxWhJnhgLMJ>. Retrieved 08-jan-2019.

- [16] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. Wiley - IEEE, Wiley, 2014.
- [17] S. Küchler, J. Eberharder, K. Langer, K. Schneider, and O. Sawodny, “Heave motion estimation of a vessel using acceleration measurements,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 14742 – 14747, 2011. 18th IFAC World Congress.
- [18] Flexcom, “Pierson-moskowitz wave.” [Website]. Available: http://www.mcskenny.com/support/flexcom/theory_wave_pierson.html. Retrieved 19-Sep-2018.
- [19] Wikiwaves, “Ocean-wave spectra,” December 2012. [Website]. Available: https://wikiwaves.org/Ocean-Wave_Spectra#JONSWAP_Spectrum. Retrieved 19-Sep-2018.
- [20] K. Torsethaugen, S. Haver, “Simplified double peak spectral model for ocean waves,” *International Society of Offshore and Polar Engineers*, 01 2004.
- [21] Q. Guo and Z. Xu, “Simulation of deep-water waves based on jonswap spectrum and realization by matlab,” in *2011 19th International Conference on Geoinformatics*, pp. 1–4, June 2011.
- [22] T. Perez and T. I. Fossen, “A matlab tool for parametric identification of radiation-force models of ships and offshore structures,” 2009.

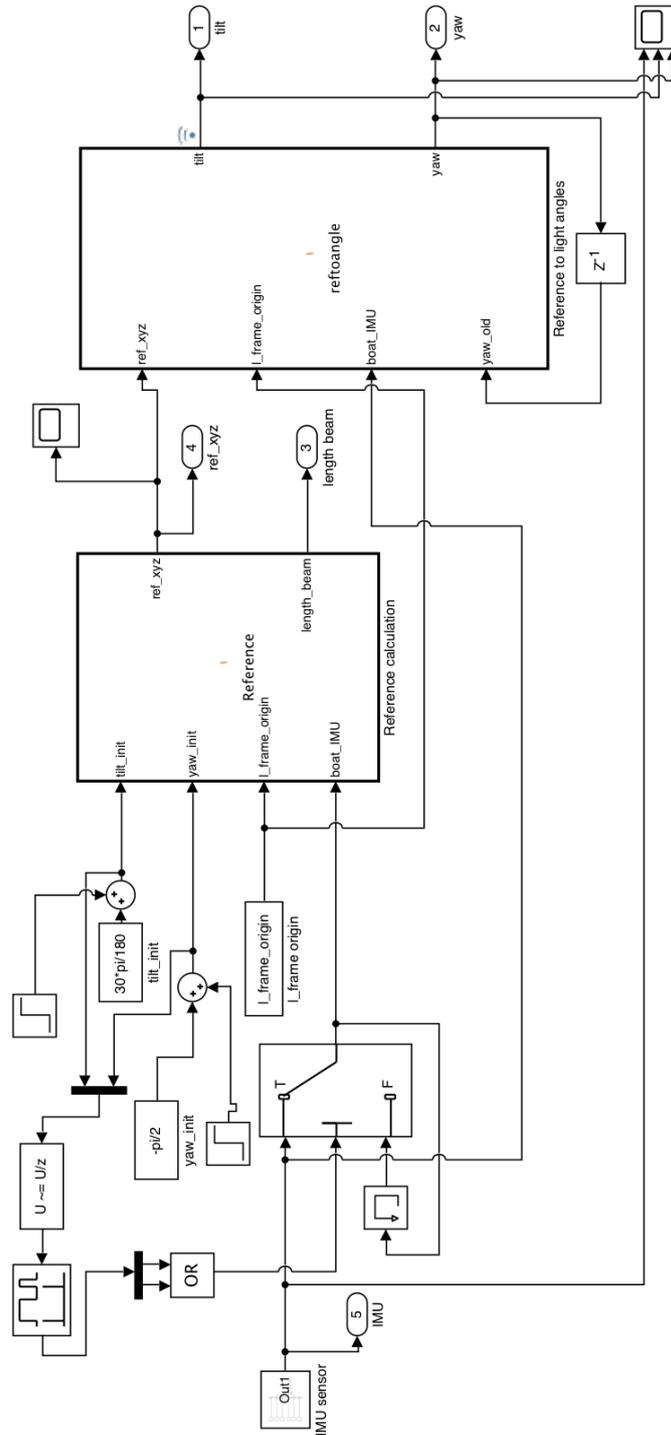
A

Appendix

A.1 Simulink model for simulation of the system



A.2 Simulink model subsystem for calculation of reference angles



A.3 Hardware used for test and verification

<https://www.clasohlson.com/se/Nyckelring-med-laserpekpinne/40-7967>

Amount	Component	Description	Link
1	IMU	MPU9150	SparkFun IMU Breakout
2	Servo Motor	SG90	SG90 DATA SHEET
1	Pan/Tilt kit	-	Pan/Tilt kit
1	Microcontroller board	Arduino Uno	ARDUINO UNO REV3
1	Microcontroller board	Funduino Mega	Funduino MEGA 2560 R3
1	Single-board computer	Raspberry Pi 3B	Raspberry Pi 3 Model B
1	Laser Pointer	-	Laser Pointer

A.4 Determining light parameters

The moment of inertia for both degrees of freedom were calculated with the formula seen in equation (A.1)

$$J = \frac{1}{2}MR^2 = 0.01 \text{ kgm}^2; \quad (\text{A.1})$$

where the mass, M , was assumed to be 1kg , and the radius, R , was assumed to be 10cm .

The parameters for the electric motors were found in data sheet for motors handling the same calculated moment of inertia together with a chosen rotational acceleration, seen in [Data sheet](#)¹ for moc23series.

$$K_m = 0.45 \frac{\text{Nm}}{\sqrt{\text{W}}}; R = 0.6 \Omega; L = 2 * 1e - 3 \text{ mH}; J = 0.01 \text{ kgm}^2; b = 0.01 \frac{\text{Nm}}{\text{KRPM}}; \quad (\text{A.2})$$

$$K_u = 0.35 \text{ Vs}; \quad (\text{A.3})$$

¹<http://www.moog.com/literature/MCG/moc23series.pdf>