NTNU

Norwegian University of
Science and Technology

Marine Inverted Pendulum

Rotem Sharoni

Marine Technology
Submission date: June 2016
Supervisor: Roger Skjetne, IMT
Co-supervisor: Andreas Reason Dahl, IMT

Norwegian University of Science and Technology
Department of Marine Technology

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MSC THESIS DESCRIPTION SHEET

Name of the candidate: Rotem Sharoni

Field of study: Marine control engineering

Thesis title (Norwegian): Marin invertert pendulum

Thesis title (English): Marine Inverted Pendulum

Background
A highly maneuverable multi-purpose marine model platform, called the "C/S Saucer", has been developed in spring 2015 in the NTNU Marine Cybernetics Laboratory (MC-Lab). The intended use of this multi-purpose vehicle is for students to design, implement, and test a variety of nonlinear guidance, control, and estimation algorithms for specified experimental case studies.

The objective of this thesis is to perform modeling and control design, simulation studies, and physical testing for balancing an inverted pendulum on a hydrodynamic platform actuated by azimuth thrusters. We believe this will be a first time ever achieved, if successful. The tasks are to study relevant background literature, to provide a dynamic model of the coupled pendulum dynamics with the hydrodynamics of the vessel, and to design control laws for the automatic balancing objective. The resulting performance shall be simulated and experimentally tested.

Work description
1) Perform a background and literature review to provide information and relevant references on:
   - MC-Lab and the C/S Saucer model.
   - Relevant inverted pendulum problems reported in the literature, including details on control designs.

   Write a list with abbreviations and definitions of terms, explaining relevant concepts related to the background study and project task.

2) Design and build an inverted pendulum payload system, according to preparations done in the project fall 2015. This includes necessary instrumentation for the C/S Saucer vessel.

3) Modify the control system on the C/S Saucer and interface necessary new sensors, such as rpm measurements, to improve the control system.

4) Derive a dynamic design model for the inverted pendulum mounted on the C/S Saucer, to be used for model-based observer and control design. Simulate responses of the model to various inputs to verify its correctness.

5) Develop an observer for the model that is sufficiently fast at estimating necessary system states. Simulate the observer's performance based on the system model.

6) Develop a control algorithm for balancing the inverted pendulum. The control objective is mainly to balance the inverted pendulum, while position control of the vessel is less important. Simulate the resulting responses to verify the effectiveness of the control law.

7) Implement the control algorithm on the real model of C/S Saucer and test the observer and control law in the MC-Lab. Present and discuss the results.

Tentatively:

8) Consider also controlling the vessel's position, while balancing the pendulum.

9) Design and implement control laws based on several control methods, for balancing the inverted pendulum. Compare and contrast their results.

Guidelines

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present his personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to 1) the main supervisor and 2) the archive, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must be submitted according to corresponding NTNU procedures. Computer code, pictures, videos, data series, and a PDF version of the report shall be included electronically with all submitted versions.

Start date: 15th January, 2016 **Due date:** As specified by the administration.

Supervisor: Professor Roger Skjetne
Co-advisor(s): PhD candidate Andreas R. Dahl

Trondheim, 30.05.2016

_____

Roger Skjetne
Supervisor

Preface

This masters thesis is a continuation of a project delivered in December 2015. I have learnt a great deal writing this thesis, however, not necessarily about the topics that were foreseen. The idea was to take a working vessel, and develop a new payload for it, and then focus on controller and observer design. However, the vessel itself was not functioning properly, and hence a great deal of time was spent on preparing the vessel. So, the knowledge I have gained is shifted more towards electronics and software, than on control theory.

It has been very frustrating at times. A lot of the electronics did not work, and on top of that I had issues with software not working. At times it has felt like I have been set back ten steps, for every step I took ahead. This felt especially true under the development of the mathematical model. That mentioned, it has also been rewarding. It felt especially good the first time the vessel with the pendulum was deployed, and that the controller was not too far from successfully balancing the pendulum.

Of course, a thesis like this is not the work of the author alone. I have had a lot of help, and would like to acknowledge the following persons.

First of all, I would like to thank my supervisor, Professor Roger Skjetne, for bringing the idea of a marine inverted pendulum to me, and for providing the resources needed, and guidance to complete the thesis.

I highly appreciate all the help I have received from my co-supervisor, Andreas Reason Dahl. He has made himself available whenever I needed, and followed my up on a weekly basis, as well as helping with proofreading and structure. He has also spent a great deal of time helping me developing the equations presented in Section 3.4, and should be acknowledged for the work presented.

Chief engineer at the NTNU marine structures laboratory, Emil Bratlie has been a valuable resource with the pendulum design. He helped a lot with ideas to the final design, and also made the payload system. He also kindly let me use the laboratory facilities, and provided me with tools.

Fellow student, Einar S. Ueland is also writing a thesis using the CS Saucer platform. I would like to thank him for always helping me when I have asked, especially with software related issues and the Robot Operating System (ROS). Without his help in ROS, this thesis would not have been completed. In this regard, Andreas Viggen Henriksen and Stian Skaalvik Sandøy have also been helpful.

Chief Engineer Torgeir Wahl has also been helpful with everything related to electronics, and for facilitating equipment in the MC Lab.

Thank you to Hans-Martin Heyn for lending me a computer when mine proved to be too slow. And especially for helping troubleshooting the CS Saucer and fixing problems with interference that caused a lot of headache.

Thank you to my father who patiently have helped me with proofreading.

Finally, I would like to thank my fellow students for support and encouragement, and a to send a special messages to my friends at the office "Grevlatoriet" for a great final year at NTNU.

Trondheim, June 9, 2016

Rotem Sharoni

Summary

The purpose of this thesis is to investigate whether an inverted pendulum can be balanced on a hydrodynamic platform actuated by thrusters. The vessel is the omnidirectional CS Saucer actuated by three rotating azimuth thrusters.

The vessel have been heavily modified, replacing the software control system with a Robot Operating System (ROS) based platform, using an Arduino Mega embedded circuit board, and the Raspberry Pi 2. This is a versatile platform, especially useful with MATLAB and Simulink simulations, and for implementation of additional hardware. For this project, rpm measurements from all motors have been added, along with battery voltage monitoring and encoder reading.

The encoder is used to read the position of the pendulum. A completely new payload system, interfacing the pendulum and encoder to the CS Saucer have been designed and constructed. The system is designed to only rotate in one plane, and the pendulum is restricted to move in the range $\pm 45°$.

In order to design a control system that keeps the pendulum at the upright equilibrium, a mathematical model of the system is needed. Equations for an inverted pendulum on a cart, and hydrodynamic modelling of marine vessels are presented, and then combined to find the equations for the system. The Lagrangian approach is used to derive the dynamic coupled equations for the marine inverted pendulum, where theory on the double inverted pendulum is used for comparison. It is also shown that the Denavit-Hartenberg convention used in robotics can be applied on the marine inverted pendulum to derive the equations.

The mathematical model is highly coupled and nonlinear. For controller and observer designs, however, it is convenient to work with a simplified model. Therefore, the system is linearized about the desired upright equilibrium and position, i.e. the zero state, and written on state space form. Based on this linear model, a linear quadratic regulator (LQR) feedback controller is designed.

Only the vessel position, pitch and pendulum angle are measured, while their derivatives also form part of the state. Since all six states are necessary for feedback control, a Luenberger state estimator is designed. Together with the LQR they form the linear quadratic Gaussian (LQG). The observer is then a combined filter and state estimator, able to reconstruct unmeasured states, and filter noisy measurements.

The system is tested in the MC Lab, where real conditions as unreliable sensors, signal losses and noise are present. Experiments show promising results for the LQR. Several runs are conducted, showing that both the control law, and vessel

should be suitable for the marine inverted pendulum. The objective is not completely achieved, but it is believed that it is possible to achieve it given some more work.

One of the main concerns in this regard is the thruster mapping. The thrusters are unreliable, especially in the zero thrust region, and when thrust direction is changed. Therefore, this should be the main focus area for further work. Moreover, the vessel design might not be the best for a marine inverted pendulum. It is directionally unstable, and a strong pitching effect is observed when thruster force is applied. So, while the omnidirectional property might be very suitable for a freely moving pendulum, a long and thin vessel is believed to be more suitable for the in-plane rotating pendulum.

Sammendrag

Hensikten med denne oppgaven er å undersøke om en invertert pendel kan balanseres på en hydrodynamisk plattform drevet av thrustere. Fartøyet er CS Saucer, drevet av tre roterende azimuth thrustere.

Fartøyet har blitt kraftig modifisert i arbeidet med denne oppgaven. Kontrollsystemprogramvaren har blitt erstattet med en "Robot Operating System" (ROS) basert plattform, ved hjelp av et Arduino Mega integrert kretskortet, og Raspberry Pi 2. Dette er en allsidig plattform, spesielt nyttig med MATLAB og Simulink simuleringer , og for integrering av ekstra maskinvare. For dette prosjektet, har turtallsmåling fra alle motorer blitt lagt til, samt spenningsovervåking av batteriet, og enkoderavlesning.

Enkoderen brukes til å lese posisjonen til pendelen. Den er del av et nytt nyttelastsystem som er designed, hvor pendelen og enkoderen kobles sammen med CS Saucer. Systemet er utviklet slik at pendelen kun kan rotere i ett plan, og den er begrenset til å bevege seg i området $\pm 45°$.

For å kunne utforme et styresystem som holder pendelen i oppreist posisjon, er det behov for en matematisk modell av systemet. Ligninger for omvendt pendel på en vogn, og hydrodynamisk modellering av marine fartøyer presenteres, og blir deretter kombinert for å finne ligningene for systemet. Den Lagrangske tilnærming blir brukt for å utlede de dynamisk koblede ligningene for den marine inverterte pendelen, hvor teori på dobbelt invertert pendel er brukt som sammenligningsgrunnlag. Det er også vist hvordan Denavit-Hartenberg konvensjonen som brukes i robotikk kan anvendes på den marine inverterte pendelen for å utlede ligningene.

Den matematiske modellen er sterkt koblet og ulineær. For kontroll- og estimatordesign, er det praktisk å arbeide med en forenklet modell. Derfor er systemet linearisert om ønsket oppreist likevekt og posisjon, dvs. null-tilstanden, og satt på tilstandsromform. Basert på denne lineære modellen, er en lineær kvadratisk regulator (LQR) tilbakekoblingskontroller utformet.

Kun fartøyets posisjon, stamp- og pendelvinkel måles, mens deres deriverte inngår også i tilstandsvektoren. Siden alle seks tilstander behøves for tilbakekoblingskontrollen, er en Luenberger tilstandsestimator utformet. Sammen med regulatoren danner de et lineært kvadratiske Gauss (LQG) system. Observatøren er da et kombinert filter og tilstandsestimator, som både er i stand til å rekonstruere umålte tilstander, og filtrere målestøy.

Systemet er testet i Marin Kybernetikk Laboratoriet, hvor virkelige betingelser

som upålitelige sensorer, signaltap og støy er tilstede. Eksperimenter viser lovende resultater for det LQR regulerte systemet. Flere forsøk viser at både kontrolloven, og fartøyet er egnet for den marine inverterte pendelen. Målet er ikke helt oppnåd i denne oppgaven, men det antaes å være mulig, gitt noe mer arbeid.

Et av hovedproblemene synes å være avbildingen fra kommandert thrust til elektrisk signal. Thrusterene er upålitelige, spesielt i regionen rundt null kraft, og når kraftretningen endres. Derfor bør dette være det viktigste satsingsområdet for videre arbeid. Det ser også ut til at fartøyets design ikke er optimalt for den marine inverterte pendelen. Det er retningsustabilt, og det er en sterk stampeffekt når kraft fra thrusterene påføres. Så, selv om den runde formen kan være svært egnet for en frittstående pendel, er et langt og smalt fartøy antatt å være mer egnet for denne oppgaven, hvor pendelen kun er bevegelig i ett plan.

Contents

List of Figures

List of Tables

List of Abbreviations

CS: "Cybership". Ship prefix used for model vessels in the MC Lab.

DIPC: Double inverted pendulum on a cart.

DOF: Degrees of freedom. Refers to number of translations and rotations of an object.

DH: Denavit-Hartenberg. A rotation and translation convention used in robot kinematics.

ESC: Electronic speed controller. Electronic unit using an electronic signal and a supply current to control the speed of electric motors.

GNSS: Global navigation satellite system. Collective term describing all global positioning systems based on satellites. Examples are the American GPS, Russian GLONAS, European Galileo and the Chinese BeiDou system.

IO: Input - Output. Connections between electronic devices in computer communication.

IR: Infrared. Light with specific wave lengths, longer than visible light, that are invisible to the human eye.

LQG: Linear quadratic Gaussian. A control system design where a Kalman filter is used together with an LQR controller.

LQR: Linear quadratic regulator. Control algorithm where a quadratic cost function is minimized to find a gain for state feedback for a linear system.

MC Lab: Marine Cybernetics Laboratory. Test basin with wave generator, towing cart and a camera based positioning system.

MIP: Marine inverted pendulum. The system described in this thesis consisting of an inverted pendulum on a free floating marine vessel.

MRAS: Model reference adaptive system. Feedback control that adapts a model to a reference system.

NI: National Instruments. See the reference National Instruments (2016).

NN: Neural network. Control technique approximating functions that can depend on a large numbers of inputs, generally unknown.

NTNU: "Norges Tekniske og Naturvitenskaplige Universitet", Norwegian University of Science and Technology .

OS: Operating system. A system that manages computer hardware and software. Examples are Microsoft Windows, Apple OS X and Linux Ubuntu.

PID: Proportional-Integral-Derivative. Common control algorithm where the state error is multiplied with a proportional gain, the integrated state error with a integral gain and it's derivative with a derivative gain.

QTM: Qualisys Track Manager. Software using infra red cameras and markers to give position and orientation of objects.

ROS: Robot Operating System. Software package providing libraries and tools for developing robot applications.

RPi: Raspberry Pi. Also used is RPi2. Processor device with digital IOs able to run operating systems. RPi2 refers to the second generation.

RPM: Revolutions per minute. A number of revolutions a propeller, for instance, does in a minute.

SDRE: State-dependant Riccati equation. Extension of the LQR where the LQR problem is solved for each time step.

Nomenclature

{b}:	Body fixed reference frame. Moves and rotates with the body.
{n}:	Denotes an inertial reference frame. Global xyz used in the MC Lab.
{p}:	Pendulum fixed reference frames. Rotates with the pendulum about its hinge.
{r}:	Denotes a set of reference frames in the "robot world".
η:	Vector with the position and orientation of a body in {n}.
θ_0:	Pitch angle. Angle between the x-axis in the {n} and {b} frames, about y.
θ_1:	Angle between the pendulum and upright z-axis in the global {n}-frame.
μ:	Input to the system.
ν:	Vector of body fixed linear and angular velocities.
∇:	Displaced volume of the vessel.
ρ:	Density.
τ:	Vector of forces and moments.
ψ:	Yaw angle. Angle between the x-axis in the {n} and {b} frames, about z.
ω:	Wave encounter frequency.
B:	Buoyancy of the vessel (force).
C_{A}:	Added mass Coriolis matrix.
C_{RB}:	Rigid body Coriolis matrix.
$C(x, \dot{x})$:	Matrix associated with velocities of the MIP.
$D(\nu)$:	Damping (linear and nonlinear) matrix.
$D(x)$:	Matrix associated with accelerations of the MIP.
g:	Gravitational constant. Usually taken as 9.81 m/s^2.
$g(\eta)$:	Restoring matrix.
$\overline{GM}_L$:	Longitudinal metacentric height.
$G(x)$:	Matrix associated with position of the MIP.
H:	Vector relating thruster commands to system states.
I_0:	Moment of inertia of the vessel w.r.t. its centre of mass.
I_1:	Moment of inertia of pendulum w.r.t. its centre of mass.
l:	Length of the pendulum on the cart pendulum system.
l_0:	Offset from centre of rotation in pitch and pendulum hinge.
l_1:	Pendulum length (MIP system).
M:	Mass matrix/pitch moment.
M_{A}:	Added mass matrix.
M_{RB}:	Mass and inertia matrix for the rigid body.
$M_{\lvert q\rvert q}$:	Nonlinear damping in pitch due to pitch motion.
$M_{\dot{q}}$:	Added mass in pitch due to pitch motion.
M_q:	Linear damping in pitch due to pitch motion.
m_0:	Point mass of the pendulum on the cart-pendulum system.
m_1:	Physical mass of the vessel in the MIP system.

N:	Yaw moment.
q:	Pitch rate.
r:	Yaw rate.
u:	Velocity along the x-axis in {b}.
W:	Weight of the vessel (force).
X:	Body fixed force in surge.
x_0:	Position in the global xyz-coordinate frame.
$X_{\dot{u}}$:	Added mass in surge due to surge motion.
$X_{\|u\|u}$:	Nonlinear damping in surge due to surge motion.
X_u:	Linear damping in surge due to surge motion.
$X_{\dot{q}}$:	Added mass in surge due to pitch motion.
x_g:	Distance from {b} origin and centre of gravity along x.
$Y_{\dot{v}}$:	Added mass in sway due to sway motion.
z:	State vector.
$Z_{\dot{u}}$:	Added mass in heave due to surge motion.
$Z_{\dot{q}}$:	Added mass in heave due to pitch motion.
z_g:	Distance from {b} origin and centre of gravity along z.

Chapter 1

Introduction

"Det har jag aldrig provat förut, så det klarar jag helt säkert!"

— Pippi Långstrump

1.1 Motivation

An inverted pendulum is a system that is naturally unstable. It is therefore a widely studied example in control engineering, as it is stabilized by control. However, although control algorithms balancing the inverted pendulum on a variety of moving carts are a well studied example, it has yet, to the author's best knowledge, not been done on a marine vessel.

The problem is mainly of academic interest. Nevertheless, a control law that successfully stabilizes the highly unstable inverted pendulum on a floating vessel is probably a robust controller that can be used in other applications. This is further discussed in Section 1.4.

1.2 Background

The main focus of this thesis is inverted pendulums, and hence relevant background is presented here. However, since the experiments shall be carried out in the Marine Cybernetics Laboratory (MC Lab) using the CS Saucer, a brief introduction to the facility and vessel follows in Section 1.2.2.

1.2.1 Inverted Pendulums

The inverted pendulum is popular among students and educators to demonstrate a naturally unstable system. It is also widely used to test control algorithms and can be used to model flight of rockets and missiles (Blakelock, 1965; Lundberg and Barton, 2009). Today there are existing a huge variety of inverted pendulum systems, some depicted in Figure 1.1.

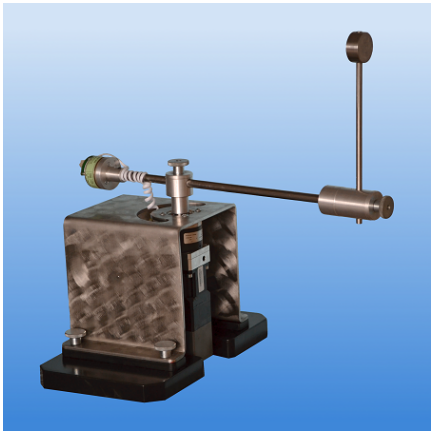
Lundberg and Barton (2009) are the source for this paragraph, summarizing the history of the inverted pendulum. They name Roberge (1960) as the first demonstrating a solution to the inverted pendulum. Three years later, in 1663 the first multiple inverted pendulum was described, while a pendulum with a vertically oscillating base was treated by several articles in 1965. The first swing-up controller for the pendulum was developed in 1975, and the rotary pendulum known as the Furuta pendulum in 1991. The latter is depicted in Figure 1.1a.

With the introduction of flying drones, it was only a matter of time before someone balanced an inverted pendulum on a quadcopter. One example is seen in Figure 1.1d where a linear quadratic regulator (LQR) control law have been applied by Hehn and D'Andrea (2011).

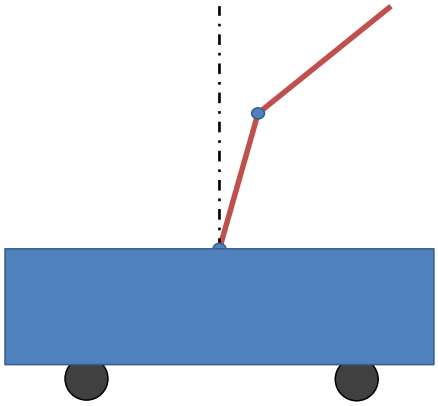
The article by Zhang et al. (2016), currently in press, should also provide some relevant background, as they propose a cascade control to balance an inverted pendulum on a quadcopter. This should be relevant to the CS Saucer-Pendulum dynamics. The article also includes a comparison with the LQR scheme widely used for inverted pendulum systems.

A more extensive background, and more examples on the variety of inverted pendulums was included by the author in the project preceding this thesis (Sharoni, 2015).

Models of inverted pendulums are well covered in the literature, starting by Roberge (1960), as mentioned. The notes by Dr. White (1997) gives a detailed example of how the inverted pendulum on a cart can be modelled by using Newton's laws. Their works, along with the control engineering books by Franklin et al. (2010)



(a) The Furuta pendulum balanced by controlling the rotation of the horizontal arm. Picture: REX Controls (2016).



(b) Double inverted pendulum on a cart.

(c) Segway. Picture: Pragues Segway (2015).

(d) D'Andrea (2013) showing an inverted pendulum balanced on a quadcopter in 6 DOF.

Figure 1.1: Examples of inverted pendulums.

and Ogata (2010) provide a good introduction on modelling, including stability analyses.

The control algorithms for the inverted pendulums vary. Some are designed to stabilize the pendulum at its upright equilibrium given it is initialized near that point. Others again can be hybrid controls where the system first has to bring the pendulum from its stable equilibrium hanging down, to the upright position. Then the stabilizing control keeps the pendulum upright.

Roberge (1960) shows in his work that a double integrator is sufficient to stabilize the inverted pendulum. However a lead network is also implemented to offset integrator lag, and to force the closed loop poles to the left half plane (Lundberg and Barton, 2009). While this stabilizes the pendulum, there is no feedback from the position. Although Roberge (1960) is not using a cart in his setup, this would cause drift of the cart. To compensate, an outer feedback loop is also added, giving position feedback.

As Lundberg and Barton (2009) write, bang-bang (on-off) controllers are also shown to work for the inverted pendulum system. However, it seems that the LQR and PID control schemes are most frequently used. Ogata (2010, pp. 756-761), for instance, uses a LQR controller with feedback from the error signal to stabilize a pendulum on a cart.

For the double inverted pendulum, Bogdanov (2004) shows, with simulations, that a controller based on a state-dependant Riccati equation (SDRE) has superior performance compared to the LQR. As they explain, the SDRE is an extension of the LQR, where the LQR problem is solved for each (discrete) time step.

Neural network (NN) control, and its combination with the SDRE and LQR are also demonstrated by Bogdanov (2004). The NN controller shows superior performance comparing to the LQR, but it has the same recovery range. In combination with the LQR (and SDRE) however, both recovery range and overall performance is increased. Nevertheless, the SDRE alone shows a pendulum recovery range 55 % to 91 % larger than the LQR.

The LQR is also used for comparison by Hassanzadeh et al. (2011). They show how an adaptive controller can be designed to overcome unknown nonlinear model parameters. Their design is based on a model reference adaptive system (MRAS). This controller uses the Lyapunov function to eliminate the defined state error, and shows superior performance over the LQR in both steady-state and disturbance response.

There are two other techniques that are applicable to the inverted pendulum system. Since the system is highly nonlinear, feedback linearization can be applied,

as it allows to cancel all the nonlinearities in the system (Fossen, 2011, Ch. 13.2). This technique is suitable for LQR controller designs, or pole placement techniques. However, feedback linearization heavily depends on that the model parameters are well known, which often is not the case.

Integrator backstepping is a technique closely related to feedback linearization, but it gives greater flexibility as nonlinearities can be kept in the controller. According to Fossen (2011, Ch. 13.3), integrator backstepping appeared in the late 80s. This approach allows the controller to be designed recursively, using Control Lyapunov functions.

The inverted pendulum balancing on a moving cart was the inspiration that led to the invention of the Segway (Franklin et al., 2010, p. 56), Figure 1.1c . Moreover, theory developed for the inverted pendulum is used to model missiles during the initial take-off phase (Blakelock, 1965). This theory also yields for aircrafts. They are both subject to aerodynamic loads, but an understanding of hydrodynamics can be modified to yield for aerodynamics, and hence this thesis might aid in that field.

A neat feature with the Segway shown in Figure 1.1c is that the mass and length of the user is a varying and unknown factor. This arises the need for a robust, and adaptive controller fitting a range of different users.

However, the marine inverted pendulum with coupled motions, remembrance a double inverted pendulum, as depicted in Figure 1.1b. Bogdanov (2004) and Zare et al. (2009) show how the mathematical of the double inverted pendulum on a cart (DIPC) model can be derived using the Lagrange equation. Then how it can be written in a compact form, where system matrices are functions of the current state. Bogdanov (2004) calls this a pseudo-linear state-dependant coefficient form.

1.2.2 NTNU MC Lab and the CS Saucer

The Marine Cybernetics Laboratory is a facility suited for testing of small marine vessels. It is a small basin with dimensions L x B x D = 40m x 6.45m x 1.5m that was originally designed as a storage tank for paraffin wax ship models (Ntnu.edu, 2015).

The laboratory is fitted with a real time positioning system both over and under the water surface. This is the Qulisys motion capture system consisting of three infrared (IR) cameras and the Qualisys Track Manager (QTM) software. The IR cameras are located as seen in Figure 1.2. In combination with IR markers

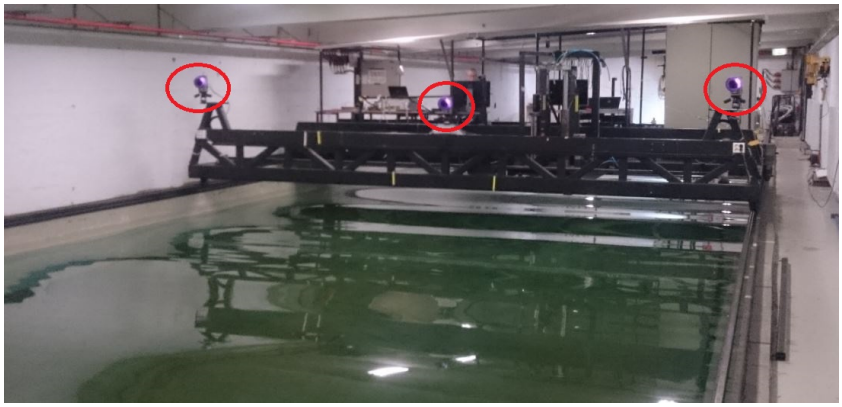


Figure 1.2: MC Lab basin with the three cameras used for surface positioning marked in red.

on the vessel, the system can calculate the position and orientation of the vessel using triangulation. This is used to emulate a GNSS system normally available in full scale. The laboratory is also equipped with a wave generator and a towing cart.

As of today, three ship models are available for use in the facilities; CS Enterprise I, Cybership II and Cybership III. The unconventional CS Saucer was introduced by Idland (2015). The laboratory also has a 1:100 model scale semi submersible drilling rig, and a turret moored drillship under development during this semester by Bjørnø (2016).

The CS Saucer (Figure 1.3) is an omnidirectional vessel (depending on the thruster configuration), hence giving the same behaviour in surge and sway, also allowing for a rapid yaw motions. These features are difficult to archive with a traditional hull shape. Hence, the CS Saucer is believed to be a suitable platform for marine inverted pendulums rotating in all directions.

Designed to be a versatile platform, the CS Saucer allows for many payload configurations and hence parameters as mass, draft etc. will change. Nevertheless, only containing the necessary hardware to run it, the mass is 3.4 kg. The diameter is d = 548 mm at the top, and d = 398 mm at the bottom.

A more extensive guide to the laboratory, as well as the mentioned vessels can be found in the laboratory handbook available at GitHub (NTNU, 2016).

In this context it is worth mentioning the aPad devolped by subCULTron (2015), that can be seen in the video by UNIZG (2016). This vessel is very similar to the CS Saucer, but seems to be more inspired by a quadcopter, as it has four symmetrically placed thrusters.

Figure 1.3: The CS Saucer.

1.3 Objectives

The inverted pendulum is, as stated, a well studied problem, and there exists a vast variety of combinations and variations of it. Nevertheless, to the authors best knowledge, there exists no evidence in the literature of a marine inverted pendulum. For a marine vessel, the pendulum motion will be coupled with the dynamic motion of the vessel itself, adding a new dimension to the world of inverted pendulums. This thesis is devoted to study the marine inverted pendulum.

The paramount objective is to successfully keep an inverted pendulum balanced at its upright position on an omnidirectional marine vessel. All experiments are conducted in the Marine Cybernetics Laboratory at NTNU. However, to successfully achieve the main objective, several partial goals are set. Hence, the thesis provides:

- Background and literature review on inverted pendulums and appropriate controller and observer designs.
- Background on the NTNU MC Lab and CS Saucer.
- Building of an inverted pendulum payload system.
- Electrical wiring and software implementations and developments on the CS Saucer.
- Derive dynamic equations for the marine inverted pendulum.
- Develop a controller based on the dynamic model for balancing the inverted pendulum.
- Design an observer based on the dynamic model.

- Simulate the different models to verify correctness, and test control and observer algorithms.
- Implement a control algorithm on the physical model of the CS Saucer and test the control law in the MC Lab.

Hence, the thesis seeks to answer whether it is possible to keep an inverted pendulum balanced upright on a marine vessel, by control. If so, will it be possible on the CS Saucer given the hardware used?

1.4 Thesis Contributions

This thesis brings a new approach to the inverted pendulum, by attaching it to a free floating vessel. Hence, this thesis contributes with:

1. Mathematical modelling of a marine inverted pendulum.
2. Development of a laboratory platform. This covers improvements and modifications to the existing platform, as well as implementation of new hardware and software. Especially, a payload system for the inverted pendulum.
3. Controller design and validation by simulations and tests in the laboratory.
4. Validation of the models by computer simulations and laboratory experiments.
5. A collection of relevant literature on inverted pendulums, from the first documented model, to advanced variations.

Furthermore, it is reasonable to believe that the findings of this thesis can give inspiration to new technology and inventions that reach beyond the aforementioned. For instance, securing buildings subject to earthquakes. Such a problem might be modelled with the building acting as a pendulum, and the earth being the hydrodynamic (moving) platform.

Offshore wind turbines for energy production is an increasingly popular field of research. As a floating wind turbine is very similar to the marine inverted pendulum, findings of this thesis might prove useful in the future of offshore wind turbines. Applications can also be extended to marine cranes.

1.5 Thesis Structure and Notation

It is frequently seen in the literature that vectors and matrices are written in a bold typeface. Nevertheless, throughout this thesis only regular style font is used for both scalars and for vectors and matrices. The dimension of each symbol is therefore clearly defined in the text. For example, $M \in \mathbb{R}^{n \times m}$ is a matrix with n rows and m columns, while $M \in \mathbb{R}$ is a scalar.

The thesis is organized as follows:

Chapter 1 supplies the reader with an introduction to the thesis. The main focus is on providing relevant background information, focusing on inverted pendulums. The objectives and contributions of this thesis are also provided.

Chapter 2 is devoted entirely to the experimental setup of the vessel and pendulum in preparation for experiments in the MC Lab. The electrical wiring of the system is presented and explained, as well as the pendulum payload. This section also introduces the software architecture designed for the CS Saucer for the project. Useful wiring diagrams and a signal flow diagram are presented here.

Chapter 3 is one of the main contributions of this thesis. The mathematical model is developed here. The section is built piecewise, presenting the pendulum and vessel dynamics separately, gradually adding complexity. Section 3.4 is the main contribution, where the dynamic equations for the combined system are derived and presented.

Chapter 4 covers controller and observer design. A linear model is presented and used as a basis to develop a LQG system. Other controllers and observers are also mentioned here as suitable for the system.

Chapter 5 is where all the results are presented. These are both simulation results, testing controllers and observers, and experimental results from the MC Lab.

Chapter 6 provides a discussion on the results presented in Chapter 5. Discussions around other issues presented in Chapter 2 and 3 are also brought up here.

Chapter 7 concludes the thesis, and presents topics considered relevant for further investigation.

Appendix A contains all the MATLAB, Simulink and Arduino files used in the project. It is therefore only included electronically.

Appendix B provides a user manual for the CS Saucer and inverted pendulum.

This includes installing and preparing the required software, and deploying and running the CS Saucer with the inverted pendulum.

Appendix C list parameter values used for simulations to verify the theory in Chapter 3.

Chapter 2

Hardware and Experimental Setup

2.1 Pendulum Payload

The first requirement for the marine inverted pendulum to be tested on the CS Saucer is first and foremost a pendulum. Then, a structure attaching the pendulum to the vessel. Several designs were proposed by Sharoni (2015). The final solution is presented in the following.

2.1.1 Design Specifications

The pendulum is designed to satisfy the following key features:

1. A Sensor measuring angle must be interfaced.
2. It must be as light as possible.
3. It must be stiff and rigid.
4. Rotation should only be allowed in one plane.
5. Possibility to alter the length and mass (including mass centre) of the pendulum.
6. Possibility to limit the angle of movement in each direction.

If property number one is not met, the control problem is not possible (Given no other measurements are added giving the orientation of the pendulum rod). Hence,

it must be possible to interface a sensor giving the angle of the pendulum relative to the upright horizontal, as the control objective is to control this angle.

The second property is important for two main reasons. First of all, the lighter the total weight of the vessel, the more responsive it is. Secondly, a heavy pendulum can cause the vessel to tip over in the water when it reaches the endpoints.

Property number three is related to modelling. The bending and flexibility of aluminium is negligible for this application, hence, rigid body dynamics can be applied to the pendulum. If the pendulum is flexible, the complexity of the mathematical problem grows massively.

The reason for adding the fourth property is that the problem is complex. Hence, by initially allowing the pendulum to rotate in one plane only, the complexity of the problem reduces, and the objective of balancing the pendulum is more likely to be achieved.

The fifth specification is desirable as it allows to examine how the length and weight of the pendulum influences the dynamic behaviour of the system, as well as choice and tuning of controllers.

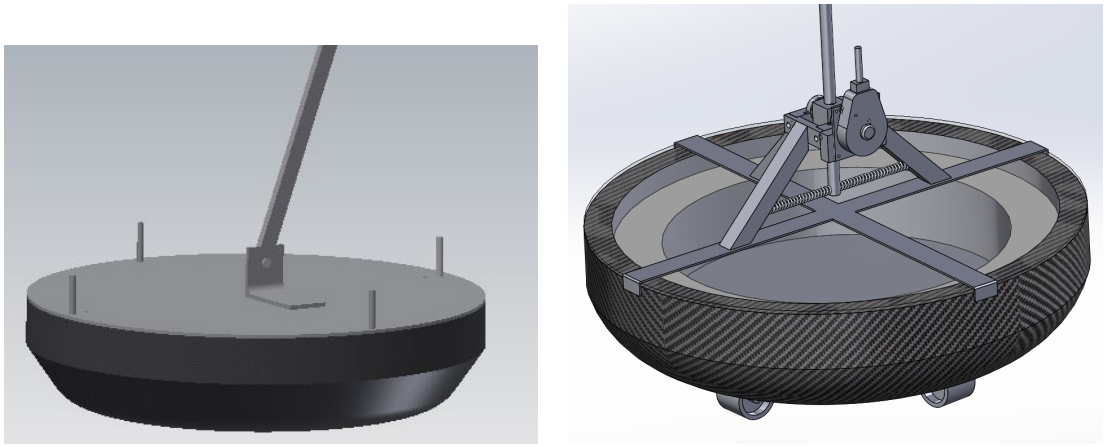
Finally, property number six is to avoid the vessel tipping over when the pendulum falls to far over. And also to avoid the rod to fall forcefully directly on the vessel hull as this may cause damage.

2.1.2 Construction

Now, it is important to keep in mind that the construction is to be made of cheap materials, mainly using left-over materials. This leaves out high end carbon designs, for instance. Therefore, scrap aluminium is utilized as it is both relatively light in weight, and stiff enough for this application.

However, a conflict between property two and three listed in Section 2.1.1 arises when it comes to the attachment between the vessel and pendulum. The author earlier proposed a simple, but lightweight design (Sharoni, 2015, Ch. 3.4) as shown in Figure 2.1a. It turned out, however, that the plexiglass lid is not stiff enough to act as a stable base for the pendulum. Therefore a supporting structure has been made as seen in Figure 2.1b.

Although this leads to a heavier structure, the weight of it, including the encoder and pendulum rod is 1.05 kg. By comparison the plexiglass lid alone weights in at 1.11 kg. Hence, this structure satisfies the third argument, while still not compromising property two significantly. Not only does it also satisfy the first,



(a) A simple pendulum design and light weight connection between pendulum and vessel

(b) Final supporting structure for the pendulum.

Figure 2.1: Pendulum support designs.

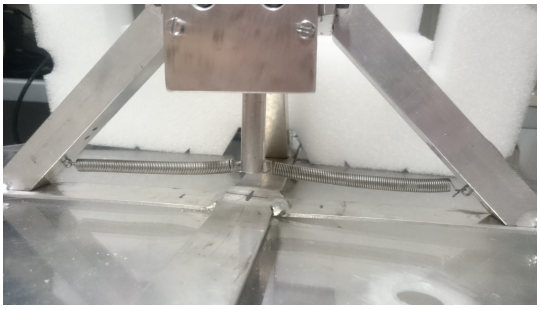
fourth and sixth property, it also allows to control the pendulum dynamics by changing the force vector from the attached springs.

The configuration in Figure 2.2a results in a relatively strong restoring force from the springs on the pendulum. This means that the pendulum will stay stable at the upright position more or less only by measure of the springs. Conversely the configuration in Figure 2.2b results in low effective force from the springs on the pendulum. This case is approaching the case without springs at all.

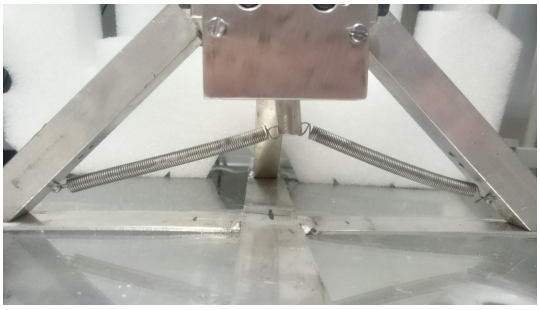
The final result is displayed in Figure 2.3b. Notice that the full length of the pendulum is not shown. Moreover, it is possible to fit additional weights along the length of the pendulum to alter the dynamics. An example is seen in Figure 2.3a, where a 35 g weight is attached.

Property five, that the length should be adjustable have not been addressed. The idea was to have a rod with slightly bigger or smaller diameter that could fit inside or outside the pendulum. That way, the length could be adjusted by drilling several holes in the pendulum, and securing it with a splint, as depicted in Figure 2.4. This has not been made due to availability of materials for the project.

To measure the angle of the pendulum an encoder has been chosen. It is rigidly attached to the supporting structure, while the encoder shaft is attached to the pendulum at the hinge. This is illustrated in Figure 2.1b. The encoder is attached with screws, and can easily be turned around, such that the wire runs down, directly through a hole in the deck.



(a) Springs more or less perpendicular to the pendulum give a relatively strong restoring force.



(b) At an angle to the pendulum the springs give less restoring force to the pendulum.

Figure 2.2: Support spring configurations.

(a) Additional weight added to the top of the pendulum.

(b) Pendulum fitted to the CS Saucer. The pendulum is longer than in the picture.

Figure 2.3: Final pendulum payload

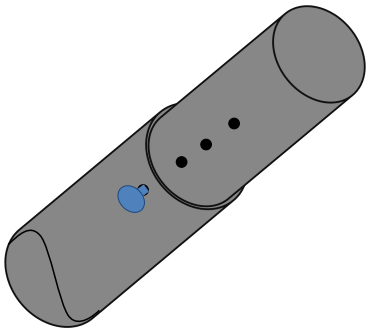


Figure 2.4: Pendulum with possibilities to adjust the length.

2.2 Control System Upgrade

Idland (2015) built and implemented a software platform based on the National Instruments (NI) LabVIEW platform. The embedded hardware device was, as mentioned, the NI myRIO. The transition to Raspberry Pi 2 allows to use the Robot Operating System (ROS), which has some advantages over LabVIEW, as discussed in the following sections.

Notice that the remaining hardware is kept as implemented by Idland (2015) and further discussed in Section 2.3. References are therefore made to Idland (2015) and National Instruments (2015) for using the "old" interface.

An overview of the new system is seen in Figure 2.5, and further discussed in the following subsections. Reference is also made to Figures 2.15, 2.17, 2.20 and 2.21 for detailed wiring diagrams.

2.2.1 Robot Operating System

The main reason for changing to the ROS platform is that Ueland (2016) uses it for interfacing the lidar. Since they also use the CS Saucer as a platform for experiments, it is advantageous to work on the same software platform. For a full introduction to ROS, readers are referred to Thomas (2014). Only a brief introduction, stating some of the advantages follows here.

ROS resembles an operating system and was released in 2007. It runs on Unix-based platforms, such as Linux. The basic idea is that the platform consist of several individual processes, called nodes. Each node performs computations and communicates with the others through messages. Each message is published on a topic, and can be subscribed to by any other node.

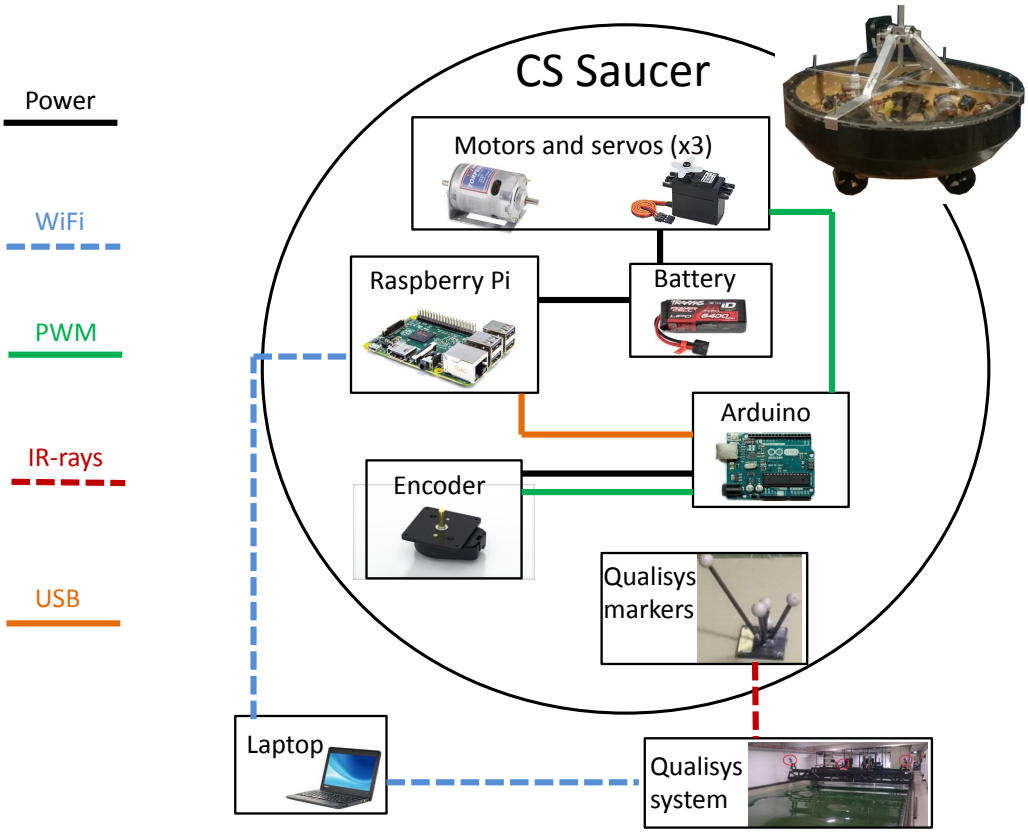


Figure 2.5: Signal flow.

One of the main advantages using ROS is that it is rapidly growing in use and popularity. Hence a lot of information is available online. As pointed out by Thomas (2014), reuse of code is the primary goal of ROS.

Furthermore, since ROS is based on individual nodes, low-level control is easily separated from high-level decision making. Hence, low-level hardware can be replaced by simulators, allowing for rapid testing. The interface also provide a simple way of recording and playing back sensor data.

For this particular thesis, a great advantage is the relative ease in which MATLAB can be used in ROS. NTNU engineering students in general have a good understanding and skill assets in MATLAB. This then saves the time and energy it requires to learn a new platform, such as LabVIEW. On the other hand, ROS itself is an extensive platform, requiring some effort and time to grasp and understand.

Figure 2.6 shows how the topics and nodes are set up in this project. Now, all topics published on the ROS network can be subscribed to by any node. That means that all topics can be subscribed to by the MATLAB node, which is the case in this project. Nevertheless, the supported code generation in Simulink means that the ROS controller easily could be transferred to the Raspberry Pi 2 (RPi2), thus eliminating the need for a laptop/computer. This means that the MATLAB and Qualisys nodes shown on the laptop in Figure 2.6, could be run from the RPi2. However, for convenience of rapidly changing parameters, and manual switching, the controller was run on the laptop for this project.

Q2E is a node converting from quaternions to Euler angles. This is since the Qualisys system publishes in quaternions, but Euler angles are more convenient to work with as they have a physical interpretation. BatterEncoderThrustRPM is a topic containing the battery voltage, motor rpms and encoder angle. This, along with the rest of the system is further explained in the following sections.

Now, Figure 2.6 shows how the communication works on the ROS framework, with the names given to the nodes and topics. This figure, and some of the names can be somewhat hard to understand. Therefore, the signal flow and communication used throughout this project is illustrated in Figure 2.7. The topics are given descriptive names now. The actual name of each node is given in Figure 2.6. The serial node is the Arduino code generated on the Arduino Mega circuit board.

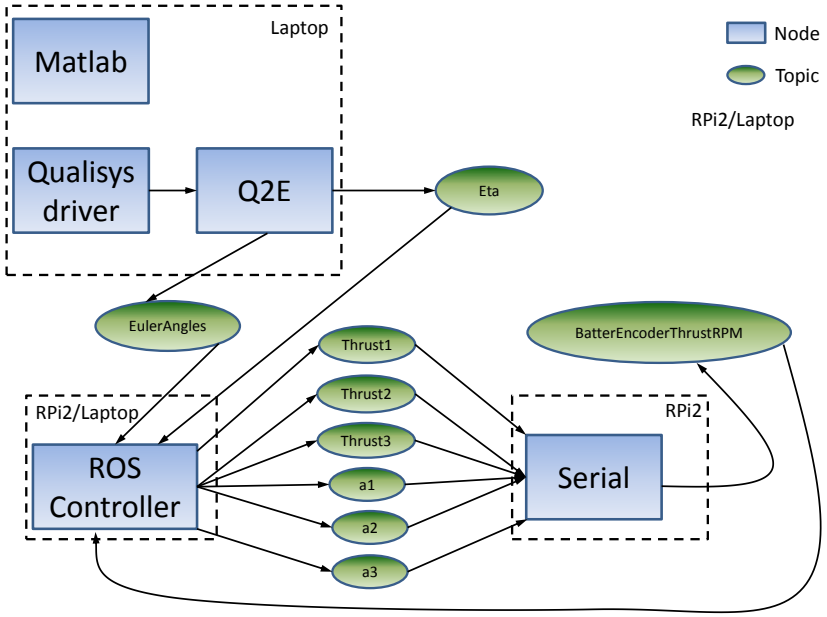


Figure 2.6: Communication in the ROS network.

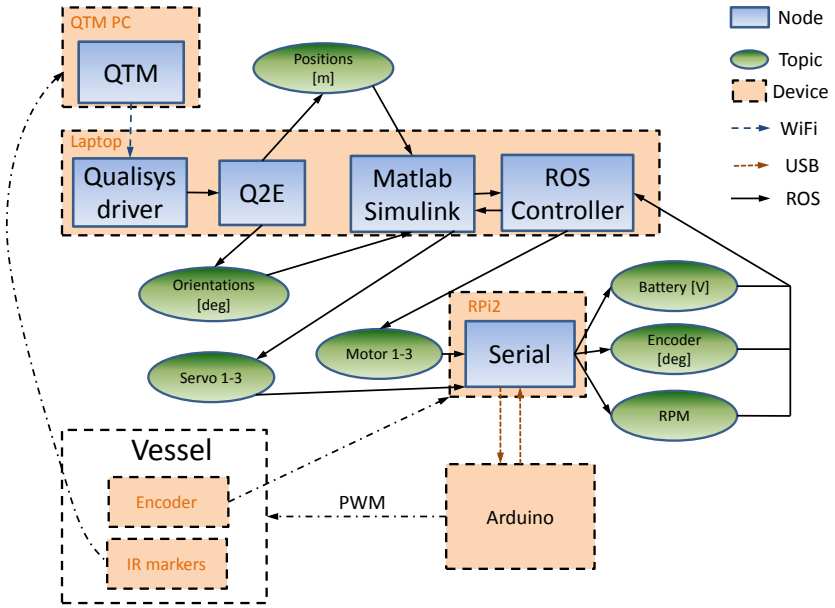


Figure 2.7: Flow of signals and communication.

2.2.2 Raspberry Pi 2

The Raspberry Pi 2 has a quad-core processor and is therefore able to run a full range of operating systems (Raspberry Pi Foundation, 2016). In this project the Linux distribution, Ubuntu 14.04.4 is installed, running ROS. This device is then responsible for all onboard computations, thus making the CS Saucer completely independent of an external computer. However, altough all code can be uploaded and run from the RPi2, it is convenient to let the user have control from an external laptop.

The Raspberry Pi 2 is connected to the local WiFi via a wireless USB adapter, and to the Arduino (Section 2.2.3) via a USB cable.

2.2.3 Arduino

The Arduino is used for low-level control of the servos and motors on the vessel, since it can guarantee real-time performance (opposed to the RPi2, which is dependant of the operating system). It is also used to interface the encoder and monitor the battery voltage, as well as the rpm of the motors.

Arduino are low cost embedded circuit boards, and a wide range of open source code and drivers are available. It is also easy to interface in the CS Saucer since it can be powered directly from the 12 V battery (Arduino, 2016), or even simpler, directly from the USB connection to the RPi2.

The Arduino operates as a separate node on the ROS framework, publishing the encoder values, motor rpm and battery voltage to ROS. On low level, PWM signals generated by the Arduino are used to control the electronic speed controllers ESC and servos. One connection is used for each servo and for each ESC, as illustrated by Figure 2.17.

Two of the digital IO of the Arduino UNO are interrupt pins. Interrupts are used to monitor events that can happen very rapidly. An especially good example is the encoder, where each pulse must be captured. This is hard to code in a traditional way, as the program would need to constantly monitor the encoder. This is solved by using an interrupt, which can interrupt the processor at any given time, and does not need to follow the structure from top to bottom in the code.

The two interrupt pins available for the Arduino UNO are pin 2 and 3. Pin 3 is also with PWM interface, so it is used to control the servos, as seen in Figure 2.17. Both the encoder and hall effect sensors need to use an interrupt, one for each rpm

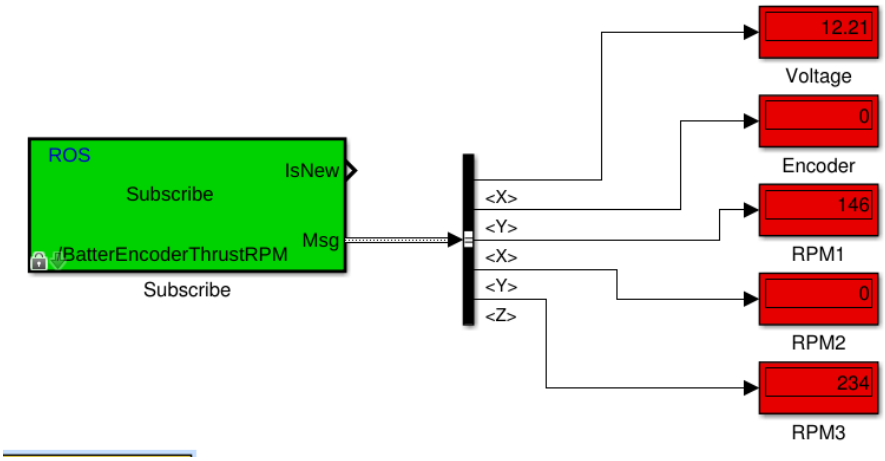


Figure 2.8: Graphical interface in Simulink. All topics published from Arduino are read in displays, or stored to data files for plotting.

sensor. Since pin 2 is the only one available, the rpm measurements and encoder can not be used simultaneously with the Arduino UNO.

Hence, two different codes are programmed in the Arduino IDE interface. The embedded circuit board must then be loaded with the code depending on whether one wants the encoder or rpm measurements. Both are attached in Appendix A.19.

Due to the above mentioned issues, the Arduino UNO, initially used for this project has been replaced with the Arduino Mega, which has six pins available for interrupts. This allows to use the encoder, battery monitor and all three hall effect sensors simultaneously. The implementation is attached in Appendix A.20.

The battery voltage level and rpm sensors are read by the Arduino Mega and published as a separate node on the ROS framework. The topics published on this node are read in Simulink as shown in Figure 2.8.

The values are post processed and used in the Simulink code, and also stored in data files for plotting.

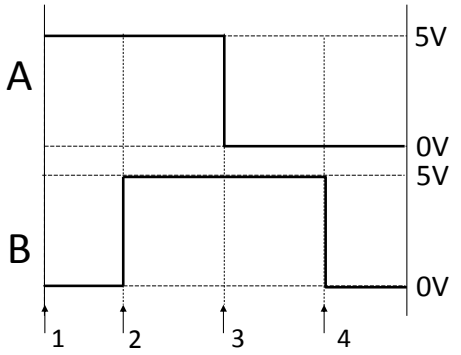


Figure 2.9: The two signal channels of the encoder during one period. The four events for each period are marked with arrows.

2.2.4 Encoder

The theory behind the encoder is widely covered online, and also briefly discussed by Sharoni (2015). Therefore the details are not explained here. In summary, the encoder has two channels, A and B, each sending a digital signal as the encoder shaft rotates. These signals are 90° out of phase, as depicted in Figure 2.9. The encoder shaft speed is determined by how rapidly each pulse is arriving. The direction of rotation can be determined by examining if channel A rises prior to B, or vice versa.

Notice that the description is valid for a single ended encoder. The US Digital (2015) encoder used in this thesis is differential, and hence allows for a much greater precision. Nevertheless, the single ended option is well within the precision required.

In the Arduino code reading the encoder (see Appendix A.19 and A.20) the properties just described are coded into the software program determining the encoder direction and velocity. The value is then published on the ROS framework, and post processed in MATLAB and Simulink.

The encoder counts 2048 counts per channel per revolution, and hence a sample reading can look like Figure 2.10 when the encoder can spin freely in both directions. As seen from the Figure, there is a jump at the wrap from 4096 to zero, and vice versa. To correct this, the unsigned data from the encoder is converted to signed data as shown in the following MATLAB algorithm:

```
function posdeg = val2deg(positionData)
%#codegen
puls = 2*2048; %2 channels x 2048 CPR
degpp = 360/puls; %Degrees pr. pulse.
% positionData = uint32(positionData); %Convert from double
    to uint32

%Convertion to signed data:
counterNBits = 12; %Bits in the encoder reading
signedThreshold = 2^(counterNBits-1); %Treshold for
   convertion to signed pos. values
signedData = positionData;
signedData(signedData > signedThreshold) = signedData(
   signedData > signedThreshold) - 2^counterNBits;

posdeg = signedData*degpp;
```

The function also converts the signed data to degrees according to the formula:

$$\text{Degrees per pulse} = \frac{360°/\text{revolution}}{2 \cdot 2048 \text{ counts/revolution}} , \tag{2.1}$$

Giving the result shown at the upper plot of Figure 2.11.

The system may be initialized at any pendulum angle within the range $[-45°, +45°]$. The final position of the pendulum in this case is the vertical upright, but the upper plot of Figure 2.11 shows a final angle of 20°.

Now, the system needs to know the exact position of the pendulum relative to a fixed vertical axis, for any initial value of the encoder. This is solved by exploiting the fact that the pendulum can travel approximately 45° to either side. The upright equilibrium is then found by moving the pendulum to the minimum and maximum deflection angle, giving

$$\text{Zero} = \text{min angle} + \frac{\text{max angle} - \text{min angle}}{2} . \tag{2.2}$$

The result is shown in the lower plot of Figure 2.11. The small bias at the end, with a final angle slightly over zero degrees is due to the fact that the travel angle is slightly bigger to one side. This is simply a bias that can be subtracted from the final value. This is done in Simulink as attached in Appendix A.16.

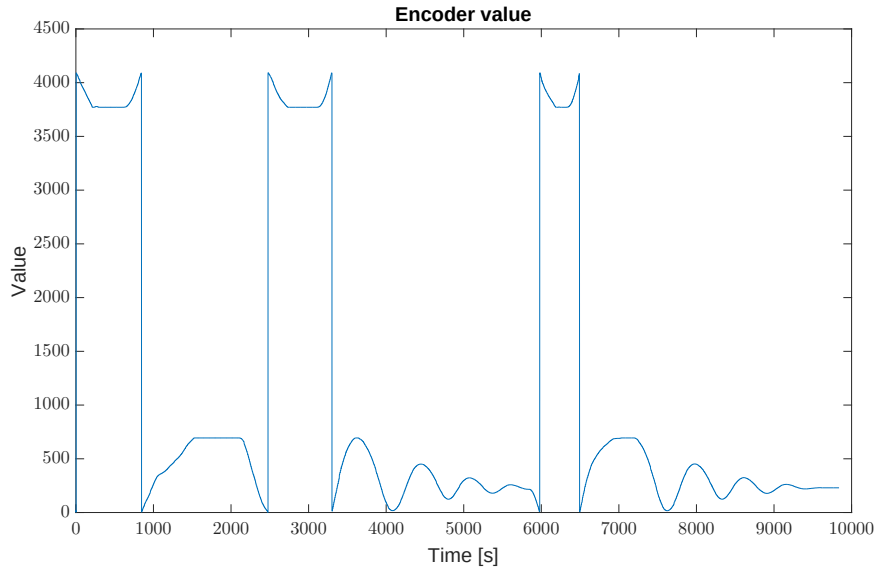


Figure 2.10: Random reading from the encoder.

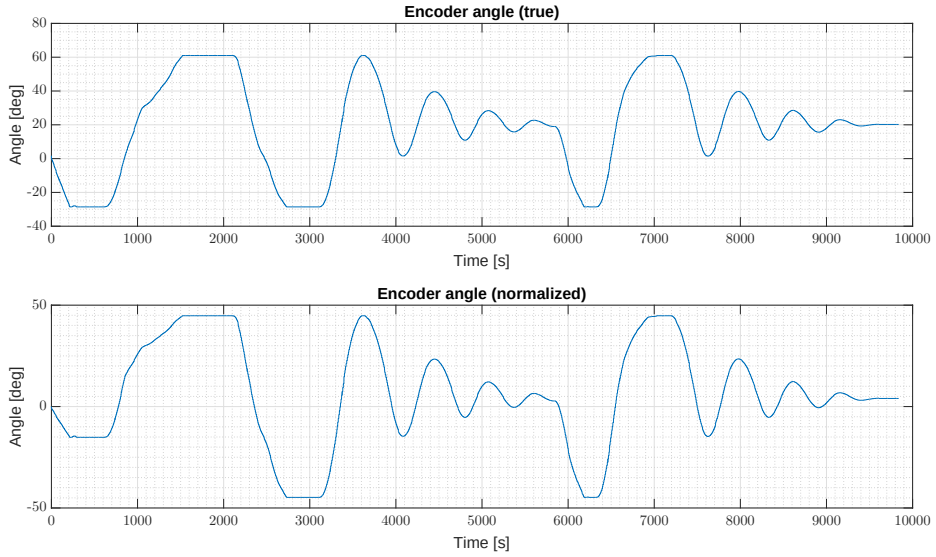
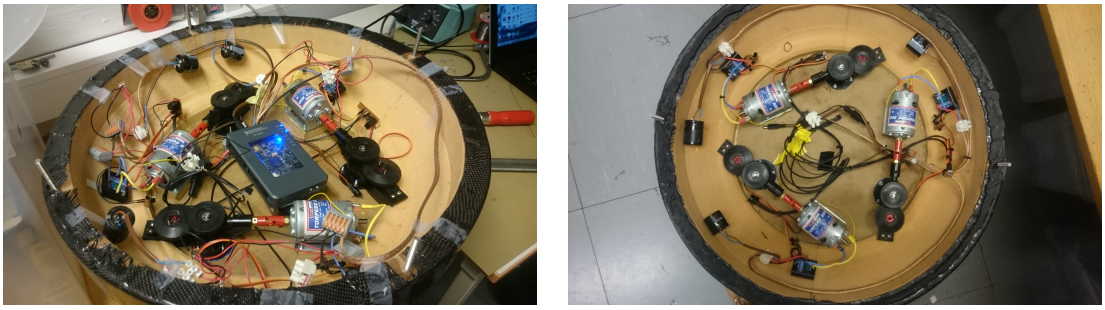


Figure 2.11: Upper: encoder reading converted to degrees. Lower: conversion to degrees, where the absolute angle is known by moving the pendulum to the upper and lower travel limits. Then the zero is found as given by (2.2).



(a) Before. (b) After.

Figure 2.12: Wiring in the CS Saucer.

2.3 Electrical Wiring and Software

As clearly seen from Figure 2.12a some preliminary work had to be done before further modifications could be implemented in the CS Saucer. The wiring was very messy, running over and in between the servo and motor gears, possibly getting stuck in the gears during operation. Furthermore, wires were badly terminated and all soldering was done without heat shrink tube to protect from short circuits.

Figure 2.12b shows the result. The wiring is now kept at a minimum, and stuck along the sides of the hull as far as possible. This minimizes the risk of wires getting stuck in gears. Furthermore it is far easier to implement new sensors and to troubleshoot the system. There is also more open space in the middle for the battery and onboard computers.

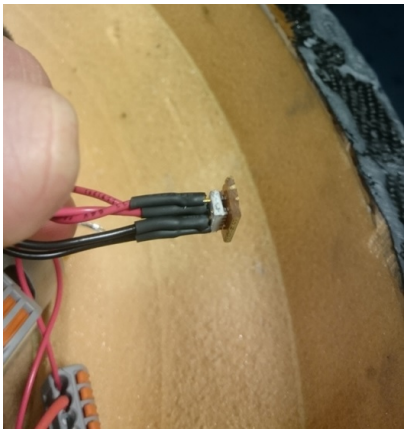
Figure 2.13 and 2.14 shows pictures before and after improvements to the wire ends have been made in form of termination sleeves and heat shrink tube. These improvements greatly reduce the risk of short circuits and moreover increases lifetime and durability of the wiring.

One of the greatest changes from the original setup by Idland (2015, Ch. 5) is that the 11.1 V to 5 V power converters have been eliminated from the circuit. This is possible since the 5 V output from the ESC to the motors have been exploited to power the servos.

The complete wiring diagram is shown in Figure 2.15. Notice that this only shows the wiring necessary to manoeuvre the vessel. External sensors, as the encoder, are not included to maintain readability.

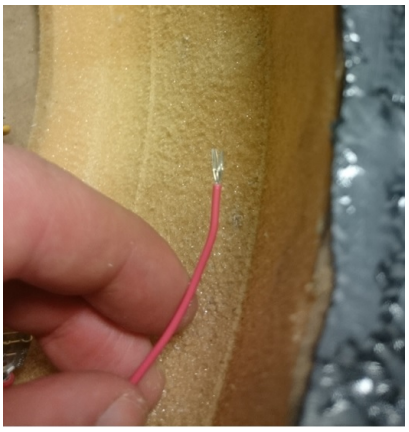
Green wires in Figure 2.15 are PWM signals, red are + 11.1 V or 5 V, while black is ground. Blue and yellow are negative and positive potential to the motors. C1,

(a) One of several soldering points with high risk of short circuit due to lacking heat shrink tube.

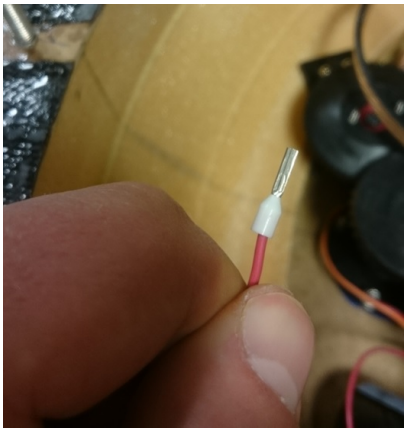


(b) Greatly reduced risk of short circuits due to isolating heat shrink on wire ends.

Figure 2.13: Electrical improvements.



(a) Wire end. The loose individual wires can break of in the terminals.



(b) Termination sleeve fitted to the wire end to ensure safe and durable termination.

Figure 2.14: Electrical improvements.

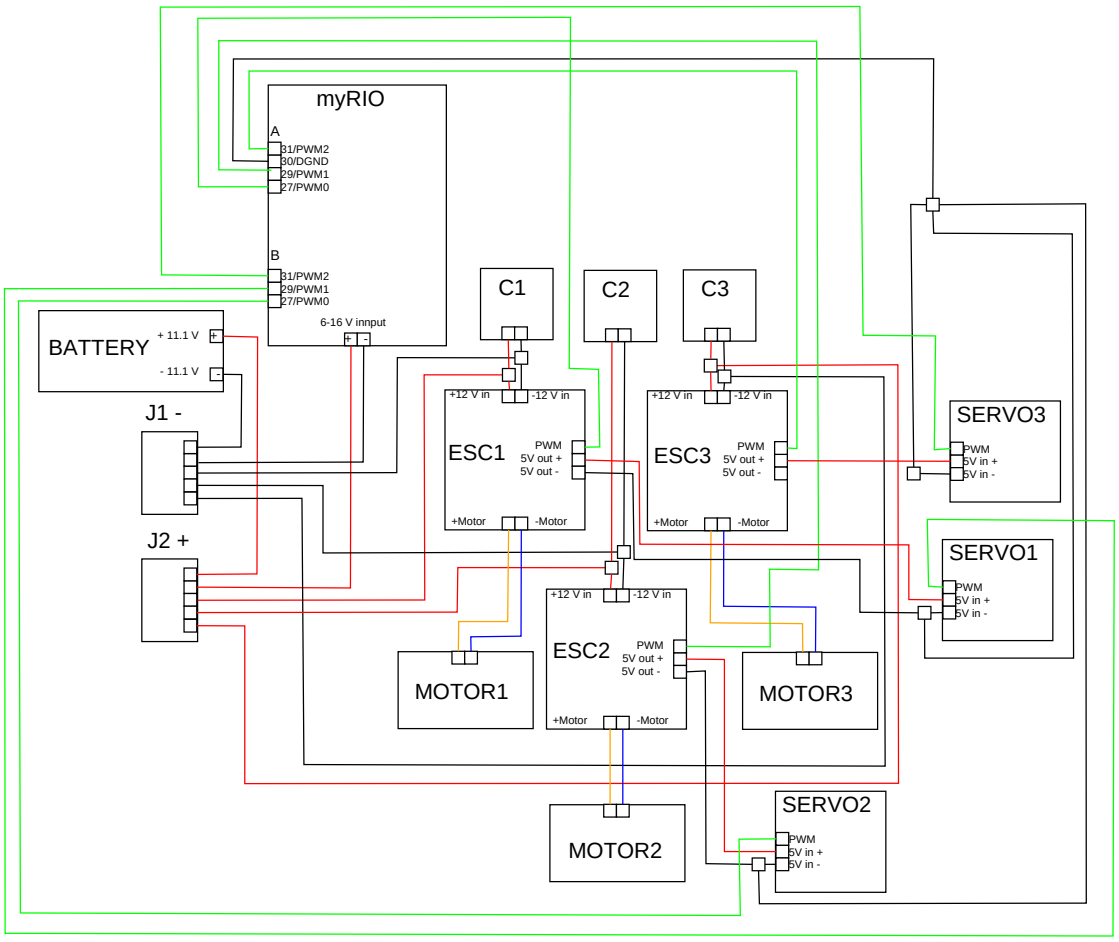


Figure 2.15: Power and actuator control signals (wiring diagram). C is capacitor.

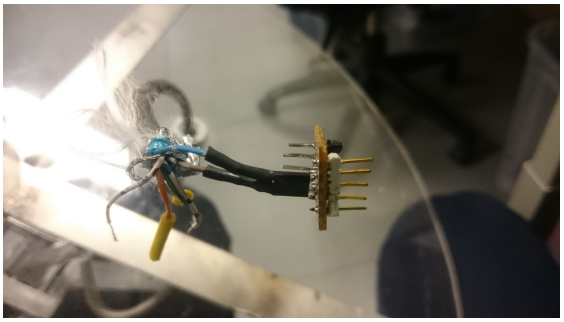


Figure 2.16: Connection of the encoder A and B channels. The black connection fits the myRIO, while an adapter is made, such that the golden pins fits the Arduino terminations.

C2 and C3 are (at least) 10000 Microfarad capacitors (Idland, 2015, Ch. 5). J1 and J2 are termination blocks where all inputs and outputs are connected.

In collaboration with Ueland (2016), the new control hierarchy has been developed. The onboard myRIO has been replaced by a Raspberry Pi 2 unit, while real time PWM signals to motors and servos are ensured with an Arduino, as mentioned. This setup is depicted in Figure 2.17. Notice that the low level wiring to ESCs and servos is kept as in Figure 2.15. The interface to myRIO is also kept, by making adapters between the myRIO interface to the Arduino, as shown in Figure 2.16.

Since all the wiring have to be kept away from water, the CS Saucer have been fitted with a silicon seal along the edge (see Figure 2.18), ensuring a waterproof seal between the lid and the hull. This required some hours sanding the edges of the vessel in order for the edge to be flat and bond properly to the sealing mass.

2.3.1 Sensors

The encoder is interfaced as a single ended type, as explained in Section 2.2.4. Moreover, the battery can be damaged if the voltage drops under 11.1 V. Therefore one of the battery cells are monitored via one of the Arduino analog inputs. The assumption is then that all three battery cells are equally charged (which they normally are), and hence the total voltage is three times the voltage in the monitored cell.

Only one cell is monitored since the Arduino only accepts up to 5 V inputs. The whole battery back can be monitored by dividing the total voltage with resistors

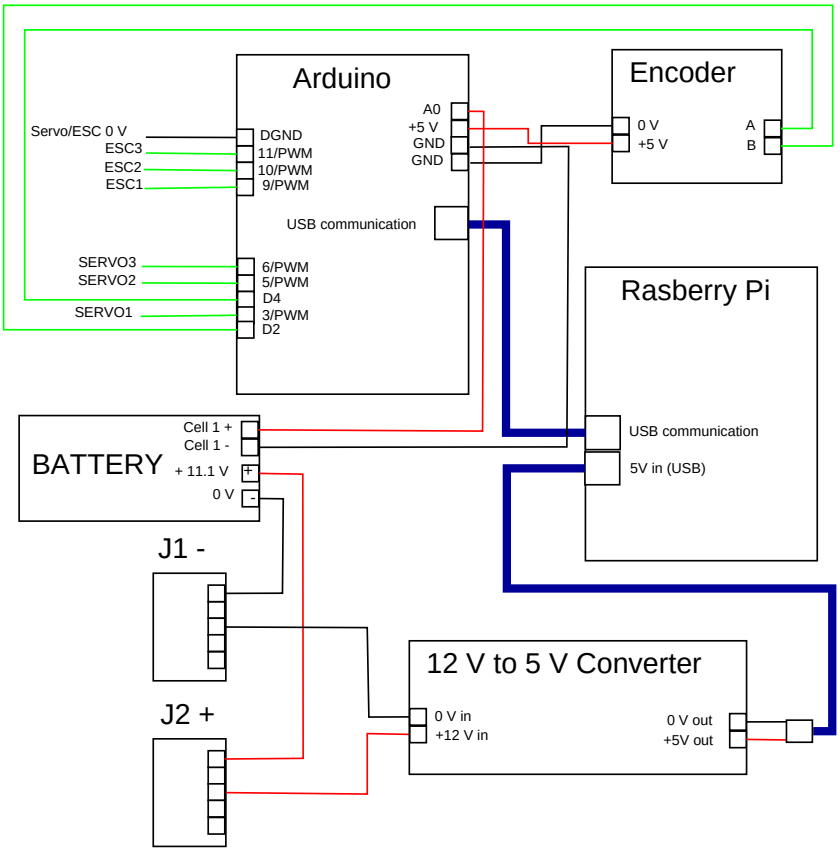


Figure 2.17: Power, actuator and control signals with RPi2 and Arduino. Wiring to motors and servos are kept as in Figure 2.15.

Figure 2.18: Seal to waterproof the vessel.

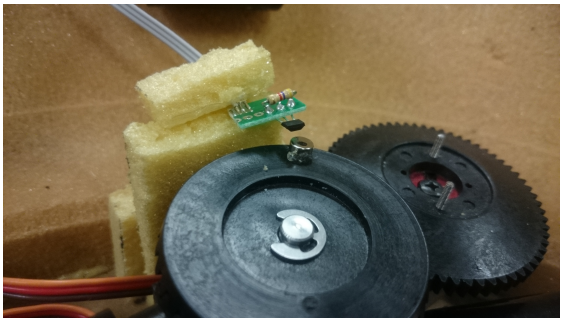


Figure 2.19: Hall effect sensor and magnet fitted to the gear of one of the motors. The actual shaft rpm is found by multiplying with the gearing ratio.

as well, but this option requires more components and work.

Another feature that has been added is rpm monitoring of the motors, enabling feedback control of the motor rpm for faster and more accurate heading and speed control. A Hall effect sensor, as seen in Figure 2.19, is used for this purpose. It is a magnetic sensor that changes its output voltage when held close to a magnet. A magnet is then fitted to the motor shaft, and registered by the sensor each time it passes. The sensor then outputs a constant 5 V, and drops to 0 V when the magnet passes, allowing to count the revolutions per minute. To find the actual rpm of the propellers, one simply has to multiply with the gearing ratio.

As explained in Section 2.2.3, the rpm sensor can not be used simultaneously as the encoder. Therefore, a wiring diagram showing the rpm sensor interface, but not the encoder, is attached in Figure 2.20.

However, since the Arduino UNO has been replaced with the Arduino Mega, as commented in Section 2.2.3, all three rpm sensors, and the encoder can be used at the same time, as shown in the wiring diagram, Figure 2.21. Notice that wiring is unchanged otherwise.

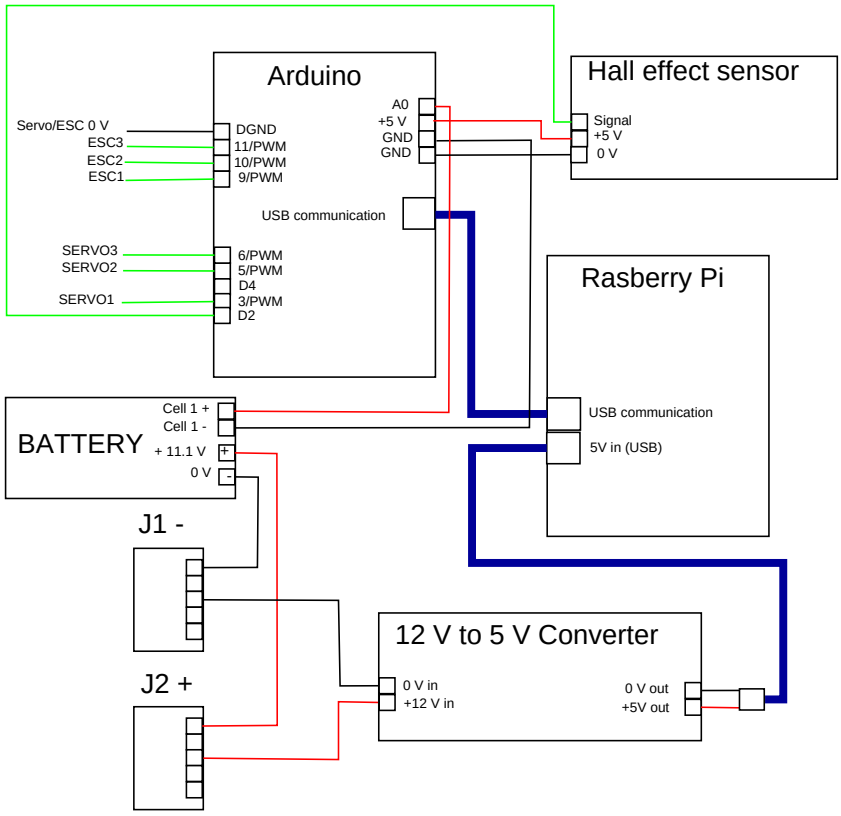


Figure 2.20: Wiring diagram where the encoder is replaced with an hall effect sensor measuring rpm. Notice that this only shows the interface for one motor. The Arduino UNO can not be used to monitor all three at the same time.

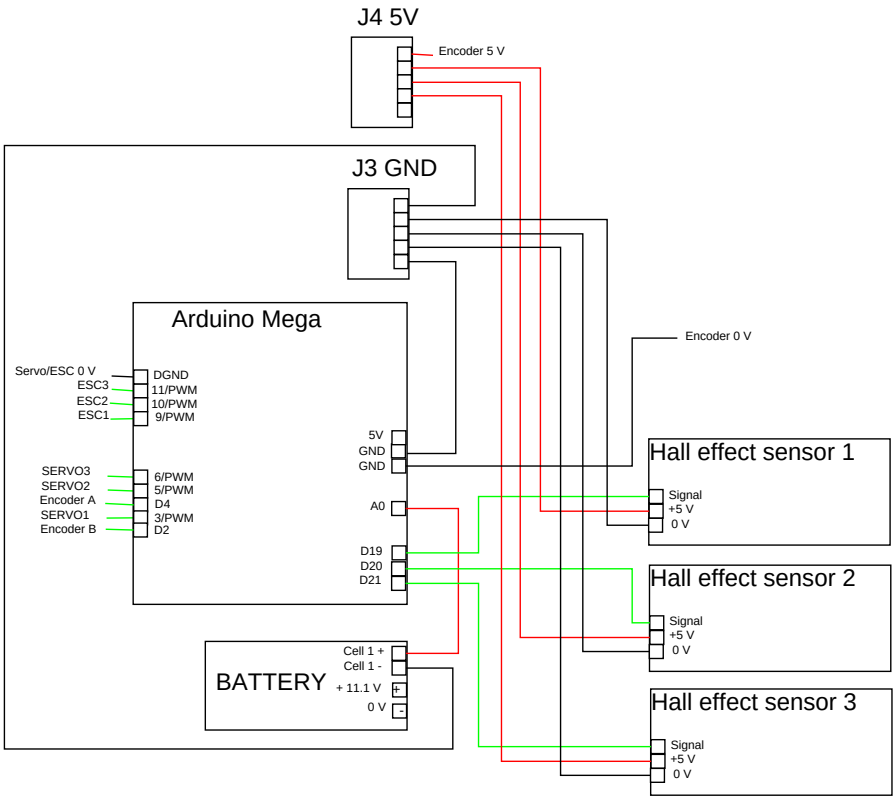


Figure 2.21: Wiring diagram for the Arduino Mega.

Chapter 3

Modelling

Balancing a pendulum on a free floating marine vessel is a complicated task. It combines hydrodynamics with control theory, where real world effects such as unmodelled thruster dynamics, noise, wave disturbances, and so on, are present. The task is therefore approached by splitting it into several parts, starting with a simple problem, and then adding complexity step by step. All steps are first verified by creating simulation models, before a complete model is developed and tested in real life.

Inverted pendulums are well described mathematically in the literature, so is hydrodynamic modelling of marine vessels. However, combining the two is, to the authors best knowledge, not been described until now. Therefore each problem is treated individually before they are combined.

Each equation is derived, and hence valid in a given reference frame. Since the task of modelling a marine inverted pendulum involves a lot of transition between coordinate frames, this chapter starts with an introduction to reference frames.

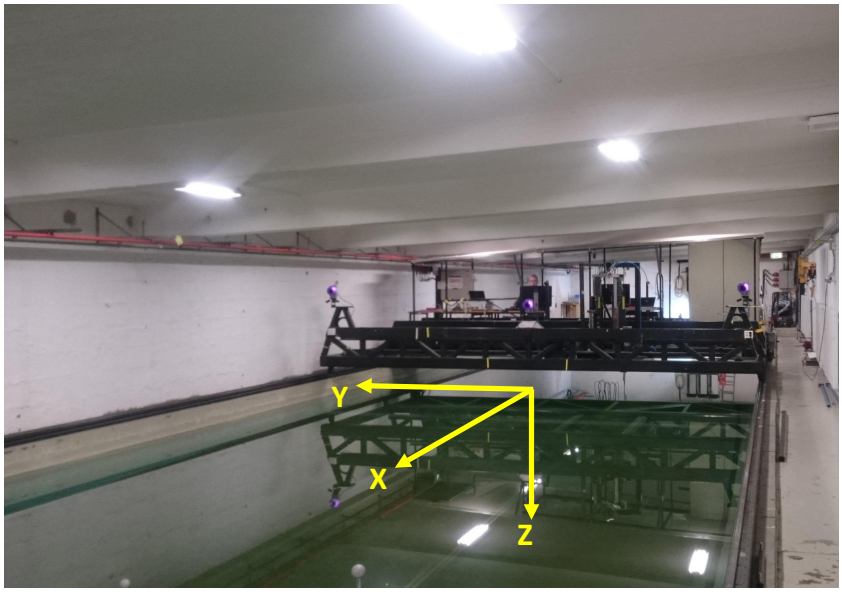


Figure 3.1: The MC Lab reference frame.

3.1 Reference Frames

In order to analyse the 6 DOF motions of the marine vessel and pendulum, some reference frames must be defined. Position and movement of the vessel and pendulum can only be defined relative to some frames.

The first reference frame used throughout this thesis is the MC Lab fixed reference frame seen in Figure 3.1. This is similar to the North-East-Down (NED) frame commonly used for local navigation in full scale, outdoors (Fossen, 2011, Ch. 2.1). The MC Lab frame is considered inertial such that Newton's laws apply. Hereafter, this frame will be denoted {n}.

Another reference frame is the body fixed frame shown in Figure 3.2. This frame is fixed to a point on the body (vessel) and hence moving and rotating along with it. This frame is from now on denoted {b}.

Notice that if the vessel is rolling, for instance, the body fixed y and z axes are rotated. To develop the dynamic equations in this frame is therefore complicated. Thus, the seakeeping ({s}) reference frame is introduced. The origin of the seakeeping frame coincides with the body fixed origin as long as the body is at rest. This is the equilibrium point. If the vessel is disturbed by, for instance, waves, such that it moves from its equilibrium, the body fixed frame moves with it, while

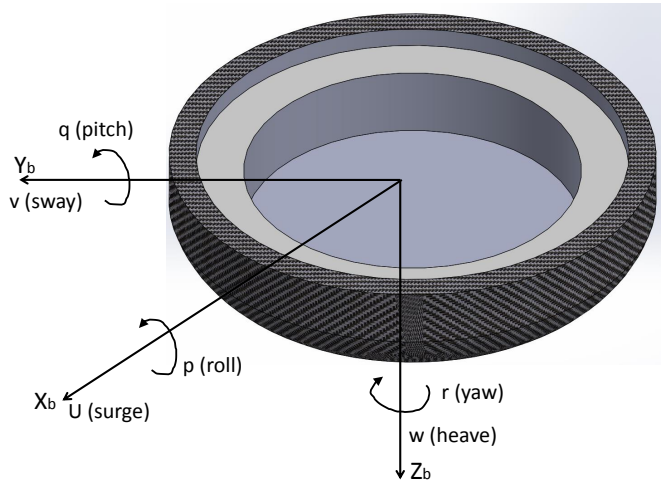


Figure 3.2: CS Saucer fixed reference frame $\{b\} = (x_b, y_b, z_b)$ and the six DOF, $\nu = [u\ v\ w\ p\ q\ r]^T$.

the seakeeping frame oscillates around the equilibrium (Fossen, 2011, Ch. 5.2). The assumption is that the {s} frame moves very slowly with respect to the {n} frame, and hence it is considered inertial.

To derive the equations of motion for the marine inverted pendulum (MIP), an additional reference frame will be introduced for clarity. This will be shown in Section 3.4, denoted as the {p} frame. Moreover, a frame set {r} is introduced in Section 3.4.3 to represent an "industrial manipulator" reference system.

3.1.1 Transformation Between Reference Frames

Different equations are derived in different coordinate frames. For instance, the vessel dynamics are normally derived in the {b} frame, while one would normally like to control the position of the vessel in the {n} frame. This topic is extensively covered in the literature, for instance by Egeland and Gravdahl (2002); Spong et al. (2006, Ch. 3); Fossen (2011, Ch. 2). Therefore, only a brief summary is given here.

Given two distinct coordinate frames, a and b as shown in Figure 3.3, the transformation from a vector given in a to a vector in b, is given as

$$v^a = R_b^a v^b \ , \tag{3.1}$$

following the notation of Fossen (2011, Eqn. 2.8). $v^a \in \mathbb{R}^3$ is a vector expressed in a, and similarly $v^b \in \mathbb{R}^3$ is a coordinate vector in b. $R_b^a \in \mathbb{R}^{3\times3}$ is then a

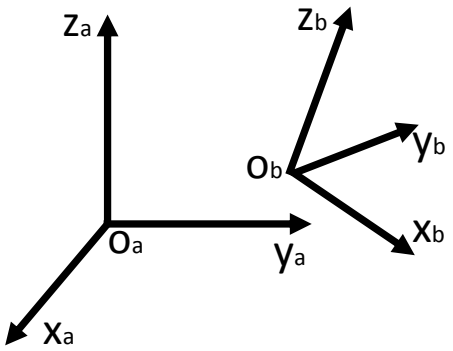


Figure 3.3: Frames a and b.

transformation matrix from a to b. If the two frames have the same origin, that is o_a coincides with o_b, R_b^a would be a pure rotation matrix, as given by Fossen (2011, Eqn. 2.18).

As will be shown in Section 3.4.3, the MIP equations can be written using homogeneous transformation matrices. This representation is inspired from robotics, allowing the different rotations and orientations of mobile frames to be expressed in a fixed reference frame.

A vector that needs N elements to be represented in $\mathbb{R}^N$ is homogeneous when represented with $N+1$ elements. In Cartesian coordinates this means that a vector in $\mathbb{R}^3$ is represented as $v = \begin{bmatrix} v_1 & v_2 & v_3 & 1 \end{bmatrix}^T$. The last entry is a scaling factor, but will always be 1 in robotics. Thus, the homogeneous transformation matrix is given according to Egeland and Gravdahl (2002, Eqn. 6.115) as

$$T_b^a = \begin{bmatrix} R_b^a & r_{ab}^a \\ 0_{1\times 3} & 1 \end{bmatrix} , \tag{3.2}$$

where R_b^a is a 3 by 3 rotation matrix, and r_{ab}^a is a 3 by 1 position vector.

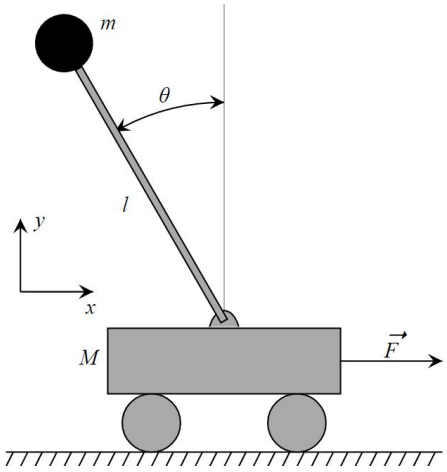


Figure 3.4: Classical pendulum on a cart moving back and forth (Krishnavedala, nd).

3.2 Inverted Pendulum on a Cart

To understand the complex dynamics of the marine inverted pendulum, the simpler problem, inverted pendulum on a cart, is studied first, as depicted by Figure 3.4. This completely excludes the hydrodynamic challenge.

The author gave a more detailed derivation of the dynamic equations for this system in the project thesis preceding this report (Sharoni, 2015, Ch. 2.2.1), resulting in a linearized dynamic equation:

$$\begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \ddot{\theta} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{(M+m)g}{Ml} & k & 0 & b \\ -\frac{mg}{M} & -k & 0 & -b \end{bmatrix} \begin{bmatrix} \theta \\ x \\ \dot{\theta} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{Ml} \\ \frac{1}{M} \end{bmatrix} \mu \ , \tag{3.3}$$

where k is some restoring coefficient and b a damping coefficient. The other symbols are defined in Figure 3.4 and in the Nomenclature.

The system (3.3) is linearized about the upright equilibrium. While this is good for regulator and observer design, the simulation model should be as detailed and exact as possible.

The linearized system (3.3) arises from the linearization of the following system,

where the restoring and damping are neglected for now.

$$\frac{d}{dt}z = \frac{d}{dt}\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \dot{x} \\ \ddot{\theta} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} z_3 \\ z_4 \\ \frac{u\cos(\theta)-(M+m)g\sin(\theta)+ml(\cos(\theta)\sin(\theta))\dot{\theta}^2}{ml\cos^2(\theta)-(M+m)l} \\ \frac{u+ml\dot{\theta}^2\sin(\theta)-mg\cos(\theta)\sin(\theta)}{M+m-m\cos^2(\theta)} \end{bmatrix} . \tag{3.4}$$

The states of the system are now collected in the 4 by 1 vector z. The equations are derived by considering force and torque balance according to Newton's second law, as shown by Sharoni (2015, Ch. 2).

As also shown by Sharoni (2015) the system has poles in the right half plane, and hence it is unstable. That means, if the pendulum is initialized at some point, not being the vertical equilibrium, it will swing back and forth, until it will stop at the downwards hanging position due to friction and air resistance.

3.2.1 Cart-Pendulum Simulation Model

The nonlinear model (3.4) has been created in Simulink as attached in Appendix A.2. This gives an exact and ideal (i.e. no disturbances) model to test control algorithms.

So, initializing the pendulum at $\theta = 20°$, one would expect to see it swinging downwards, then back and forth, never reaching the upright equilibrium. Moreover, since the model (3.4) does not include friction or drag, it will continue for all time.

This is clearly seen in the upper plot in Figure 3.5. Moreover, the rotation of the pendulum initializes the cart, making it move in positive x-direction. A visualization of the system has also been implemented, and attached in Appendix A.3. This runs a "movie" of the vessel and pendulum, where the vessel in this case is the cart. A capture from this is seen in Figure 3.6, and it verifies the assumptions discussed in Section 3.2 and the data seen in Figure 3.5.

Notice that all parameters in this simulation are randomly chosen to reasonable values, as given in Table 3.1. The dimensions are close to the CS Saucer and pendulum system, but not exact.

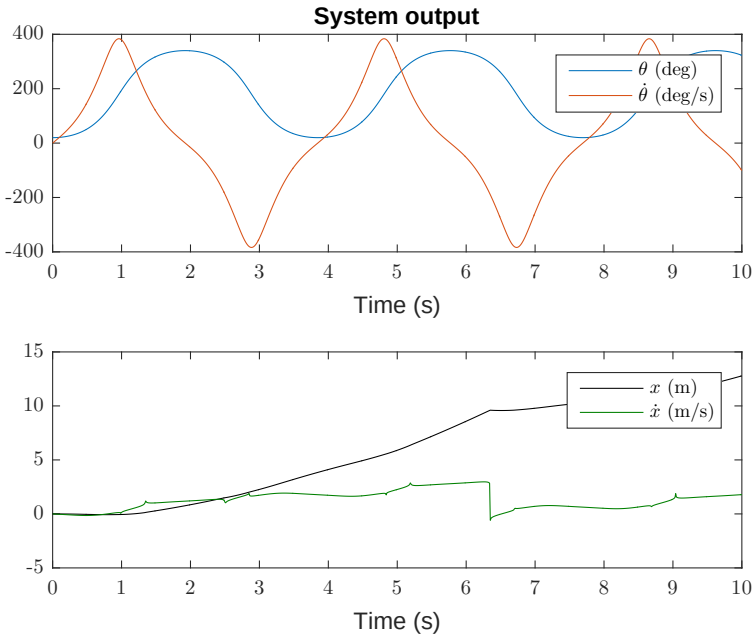


Figure 3.5: Pendulum angle and angular speed (top) and cart position and velocity (bottom), for zero input, i.e. $\mu = 0$.

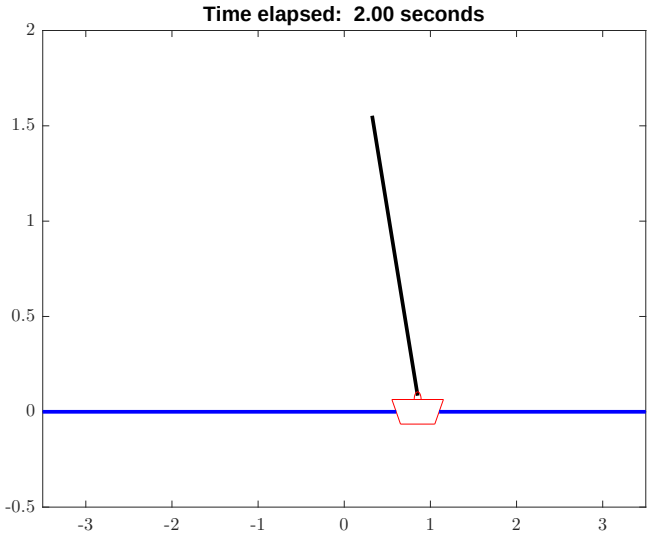


Figure 3.6: Visualization model simulating the response of the system.

Table 3.1: Parameters for inverted pendulum on a cart.

Parameter	Name	Value
Gravity	g	9.81 $\frac{\mathrm{m}}{\mathrm{s}^2}$
Mass of cart	M	5 kg
Pendulum mass	m	0.5 kg
Pendulum length	l	1 m

3.3 Vessel Equations of Motion

How to accurately describe the motions of a vessel or structure in open water is an extensive subject. This project merely touches the surface of the problem. Nevertheless, it is an important part, as the project concerns a pendulum on a free floating vessel.

Now, it is common to derive the equations by splitting up the problem into several sub-problems that are easier to solve. The hydrodynamic forces are commonly divided into two sub-problems (Faltinsen, 1990, Ch. 3):

- Forces and moments due to a rigid body introduced in a wave field. These are called Froude-Kriloff and diffraction forces and moments.
- Forces and moments due to an oscillating body in an otherwise calm sea. The body oscillates with the incident wave frequency resulting in loads identified as added mass, potential damping and restoring.

Faltinsen (1990) then shows how this can be used to express the rigid body motions of a vessel as

$$(M_{\mathrm{RB}} + M_{\mathrm{A}})\dot{\nu} + D(\nu)\nu + g(\eta) = F \ , \tag{3.5}$$

where M_{RB} and $M_{\mathrm{A}}, \in \mathbb{R}^{6\times 6}$ are the rigid body mass and inertia matrix, and the added mass matrix, respectively. $\nu \in \mathbb{R}^6$ is the six DOF linear and rotational velocities expressed in $\{\mathrm{b}\}$, $D(\nu) \in \mathbb{R}^{6\times 6}$ is the damping matrix, consisting of both linear and nonlinear damping terms. $g(\eta) \in \mathbb{R}^6$ is the vector with restoring coefficients, $\eta \in \mathbb{R}^6$ the three positions in $\{\mathrm{n}\}$, and orientation of $\{\mathrm{b}\}$ with respect to $\{\mathrm{n}\}$, while $F \in \mathbb{R}^6$ is the external forces and moments acting on the vessel. These are composed of environmental loads and thruster forces. Since this paper concerns a laboratory experiment in calm water, only the latter are included here.

Now, Fossen (2011) represents (3.5) using manoeuvring theory, also including the

Coriolis-centripetal forces due to the rotation of reference frames, giving

$$(M_{\mathrm{RB}} + M_{\mathrm{A}})\dot{\nu} + D(\nu)\nu + (C_{\mathrm{RB}}(\nu) + C_{\mathrm{A}}(\nu))\nu + g(\eta) = \tau \ , \tag{3.6}$$

where C_{RB} and $C_{\mathrm{A}}, \in \mathbb{R}^{6\times 6}$ are the rigid body, and added mass Coriolis matrices, respectively, and $\tau \in \mathbb{R}^6$ the vector of thruster forces and moments, given as

$$\tau = \begin{bmatrix} X & Y & Z & K & M & N \end{bmatrix}^T \tag{3.7}$$

in {b}.

In the notation of Fossen (2011), the full state for the vessel is given as $\eta = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$, giving the dynamic equations for the vessel expressed in body frame as

$$\begin{aligned} \dot{\eta} &= J_{\Theta}(\eta)\nu \\ \dot{\nu} &= M^{-1}\tau - M^{-1}D(\nu)\nu - M^{-1}C(\nu)\nu - M^{-1}g(\eta) \ , \end{aligned} \tag{3.8}$$

where $J_{\Theta}(\eta) \in \mathbb{R}^{6\times 6}$ is the general rotation matrix from {b} to {n}, given by (Fossen, 2011, Ch. 2.2.1). The remaining matrices are as discussed for (3.6) above, collecting the hydrodynamic and rigid body forces in one matrix for each term.

Notice that the added mass and damping terms are frequency dependant. However, it is common to use the value for $\omega \to \infty$ or $\omega \to 0$ when dealing with the equations in time domain, ω being the wave (encounter) frequency. This simplification is especially valid in the calm water in the test basin, where $\omega \to 0$ is used, as proposed by SIMO (2013, p. 28).

Now, the CS Saucer represents a unconventional ship design with both fore/aft, and starboard/port symmetry. Notice that this is not exactly true, due to the location and orientation of the thrusters. Nevertheless, neglecting this effect of the thrusters, the mass matrix in (3.6) will have the following structure (Fossen, 2011, Ch. 7.5):

$$M = (M_{\mathrm{RB}} + M_{\mathrm{A}}) = \begin{bmatrix} m_{11} & 0 & 0 & 0 & m_{15} & 0 \\ 0 & m_{22} & 0 & m_{24} & 0 & 0 \\ 0 & 0 & m_{33} & 0 & 0 & 0 \\ 0 & m_{42} & 0 & m_{44} & 0 & 0 \\ m_{51} & 0 & 0 & 0 & m_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & m_{66} \end{bmatrix} . \tag{3.9}$$

In general, it is hard to give a general structure for the remaining matrices. Fossen (2011) gives examples under certain assumptions and simplifications. Moreover

they show that there exist a variety of choices for the Coriolis matrix. Hence, it can be chosen skew symmetric, resulting in desirable properties when designing control systems and observers (Fossen, 2011, Ch. 3.3).

The hydrodynamic matrices are also hard to find in general. They are usually found by conducting a series of model tests. Skjetne et al. (2004), for instance, shows such an approach. Another approach is to use computer software for hydrodynamics, such as Wadam (DNV GL AS, 2016). Wadam can be used for arbitrary shapes, and is hence suitable for the untraditional CS Saucer. Other software, such as ShipX (Marintek (2016)) is based on strip theory, and is hence not valid for the CS Saucer. This is since strip theory is based on the assumption that the length is significantly larger than the width of the hull (Faltinsen, 1990), which clearly is not the case for the CS Saucer.

For the restoring vector $g(\eta)$, however, Fossen (2011, Ch. 4.2) shows an approximation that is valid for small roll and pitch angles, and small heave translations. All these assumptions are in general valid for the CS Saucer and pendulum system. While the pitch angle indeed can grow large due to the pendulum falling over, this will not be further covered here since the objective is to balance the pendulum upright. Hence the small angle assumption is used. This should of course be revised for more advanced tests with waves.

Equation (3.6) is the full 6 DOF motions on matrix form. However, the most relevant states for this thesis are the surge and pitch motions, and the coupling between them. These are the two main motions affecting the pendulum. Since the vessel is moving on the free surface, it is also of interest to know its position and orientation. However, these motions are assumed uncoupled from the surge and pitch dynamics.

The 6 DOF model (3.6), can be reduced to a 2 DOF model as:

$$\begin{bmatrix} m_{11} & m_{15} \\ m_{51} & m_{55} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} c_{11} & c_{15} \\ c_{51} & c_{55} \end{bmatrix} \begin{bmatrix} u \\ q \end{bmatrix} + \begin{bmatrix} d_{11} & d_{15} \\ d_{51} & d_{55} \end{bmatrix} \begin{bmatrix} u \\ q \end{bmatrix} + \begin{bmatrix} g_{11} \\ g_{55} \end{bmatrix} = \begin{bmatrix} X \\ M \end{bmatrix} . \tag{3.10}$$

The elements in each matrix is now the superposition of the physical parameters, and added mass parameters.

The corresponding forces controlling the pendulum dynamics in 2 DOF, as will be shown in the continuing, is force in surge and moment in pitch given as $\begin{bmatrix} X & M \end{bmatrix}^T = \begin{bmatrix} X & Xl_z \end{bmatrix}^T$, where l_z is the moment arm seen in Figure 3.7.

Notice that the arm is taken to the waterline, that is since the normal assumption is that the vessel will be pitching about the flotation centre (Amdahl et al., 2011, Ch. 8). That is, the centre of the water plane area seen from above.

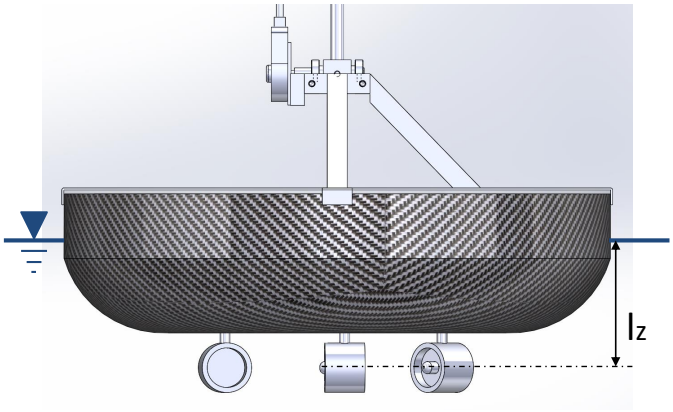


Figure 3.7: The arm from the thrusters to the waterline creates a pitching moment when force in surge is applied.

Using subscript A to denote added mass, the elements of the matrices are as follows (see also the Nomenclature):

$$
\begin{aligned}
m_{11\mathrm{A}} &= -X_{\dot{u}} \\
m_{15\mathrm{A}} &= m_{51\mathrm{A}} = -X_{\dot{q}} \\
m_{55\mathrm{A}} &= -M_{\dot{q}} \\
c_{11} &= 0 \\
c_{15} &= -c_{51} = m_0 x_g q - Z_{\dot{u}} u - Z_{\dot{q}} q \\
c_{55} &= 0 \\
d_{11} &= -X_u - X_{|u|u}|u| \\
d_{15} &= 0 \\
d_{51} &= 0 \\
d_{55} &= -M_q - M_{|q|q}|q| \\
g_{11} &= 0 \\
g_{55} &= \rho g \nabla \overline{GM}_L
\end{aligned}
$$

$I_0 = I_y$ is the moment of inertia of the vessel about the y_b-axis (pitching moment) with respect to its centre of gravity, as earlier explained.

$\overline{GM}_L$ is the longitudinal metacentric height, and is defined by for instance Fossen (2011, Ch. 4.2), and further discussed for the MIP by Sharoni (2015, Ch. 2).

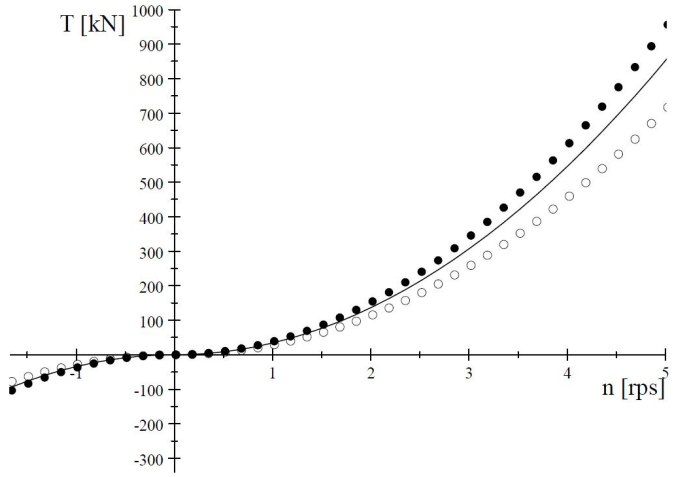


Figure 3.8: Typical example of the nonlinear relation between increasing rpm (commaned thrust) and actual delivered thrust. From Sørensen (2013).

Notice that (3.10) is given in {b}. On the other hand, the MIP equations are derived in {n}, as will be shown in Section 3.4. Therefore, (3.10) must be expressed in {n} before it can be combined with the equations for the marine pendulum.

Fossen (2011, Ch. 7.5) outlines the transformation of (3.6) into the {n} frame. This transformation also affects the system matrices. The transformation used in this thesis is given by Fossen (2011, Eqn. 7.192). The explicit expressions will not be included here, since the equation can be applied directly on (3.10).

So, the model discussed in this section captures the inertia of the vessel, separating it from the cart discussed in Section 3.2. While the cart can start, stop, and change direction at a fraction of a time, the inertia of the vessel in water will create delays.

3.3.1 Thruster Configuration

The thruster dynamics has not been modelled yet. This is the time the thrusters uses to build up momentum and force. A typical example is seen in Figure 3.8. Moreover the thrust is controlled by shaft speed, and hence also limited in both directions to a physical maximum.

A first order transfer function on the form

$$F = \frac{K}{Ts + 1}\mu \tag{3.11}$$

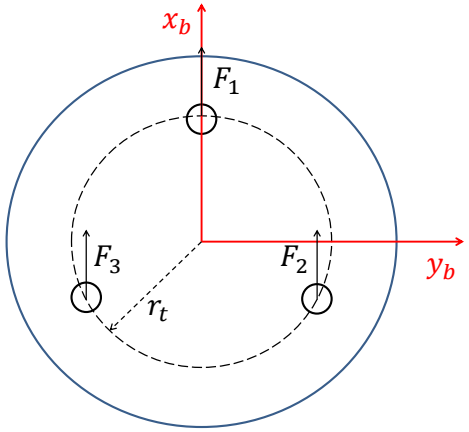


Figure 3.9: Fixed angle thruster configuration giving maximum available thrust in surge, but no actuation in sway.

can be used to model the thruster dynamics. K is some gain and T a time constant determining how fast the thrusters respond. F is the force delivered by the thruster, and μ the commanded input. This then leads to the equation for the thruster dynamics:

$$\dot{F} = \frac{K\mu - F}{T} \ . \tag{3.12}$$

Now, the vessel is over actuated, meaning that the thrust allocation problem is highly complex. However, for the purpose of this thesis, fixed thruster angles are used, resulting in a fully actuated vessel. The body fixed forces are then related to the control input as

$$\tau = T(\alpha)K\mu \ , \tag{3.13}$$

where $T(\alpha) \in \mathbb{R}^{3\times3}$ is a thrust configuration matrix, dependant on the thruster angles and location. K is a diagonal matrix, with the same dimensions as μ, relating the thruster forces to the control input vector μ.

The fixed thruster configuration is depicted in Figure 3.9. Expanding (3.13) then gives:

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & -r_t & r_t \end{bmatrix}}_{T} \underbrace{\begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}}_{F_1,F_2,F_3} \tag{3.14a}$$

and

$$M = Xl_z \ . \tag{3.14b}$$

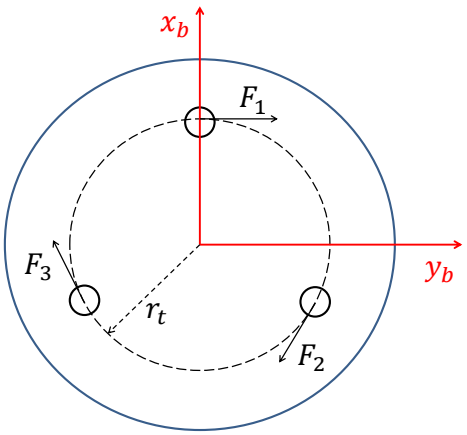


Figure 3.10: Thruster configuration. All thrusters tangential to the vessel and positive thrust defined in the clockwise direction.

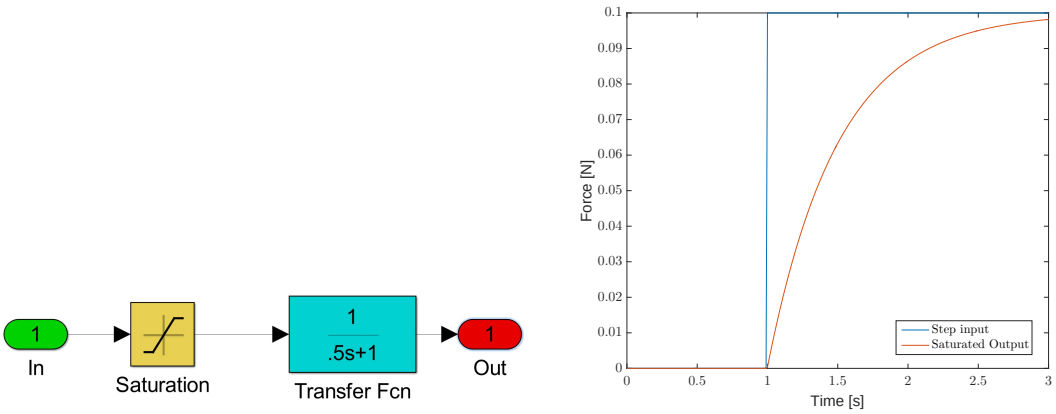
It is worth mentioning that force in surge will decrease with increasing pitch angle. In level manoeuvring, the force from the thrusters only has a horizontal component. With a pitch angle, however, more and more of this component is "lost" to a vertical component. This effect is assumed small for reasonable pitch angles, and is therefore neglected.

The configuration in Figure 3.9 is chosen as it produces the largest available thrust in surge. Since the pendulum can only rotate in the surge direction, this will give best possibility to control it. However, this configuration results in an underactuated vessel, as there is no actuation in sway.

One solution with fixed thruster angles giving a fully actuated vessel is depicted in Figure 3.10. The corresponding thrust configuration matrix is then

$$T = \begin{bmatrix} 0 & \sin(\frac{2\pi}{3}) & \sin(\frac{4\pi}{3}) \\ 1 & \cos(\frac{2\pi}{3}) & \cos(\frac{4\pi}{3}) \\ r_t & r_t & r_t \end{bmatrix} \tag{3.15}$$

Notice that with all thrusters spinning in the clockwise direction for forward speed, the vessel will drift towards starboard. To reduce this effect, the blade of thruster 3 is reversed, and the thruster spins in the opposite direction.



(a) Block diagram. (b) Step command, and thruster response.

Figure 3.11: Thruster dynamics.

3.3.2 Vessel Dynamics Simulation Model

To verify (3.8), the system has been implemented in Simulink as shown in Appendix A.4. They main interest point with this model is to check the influence force in surge have on pitch, and to model the thruster dynamics.

A first order transfer function is used to model the thruster dynamics, as seen in Figure 3.11a. A saturating element is also included to account for the physical limitations in available thrust. Notice that both the physical maximum and time delay are randomly set, as the exact parameters were unknown at the time the simulations were conducted.

Figure 3.11a shows how the thruster dynamics are modelled as a saturated delay from the commanded step input. The simulation is done in a preliminary stage, and hence the model is not mimicking the actual thrusters on the CS Saucer, as the parameters are unknown.

As mentioned, it is also of interest to investigate the influence force in surge have on pitch. To examine this, the vessel have been subjected to a step command in surge force, as seen in Figure 3.11b. This also causes a constant moment in pitch. The response is plotted in Figure 3.12.

Although the result (Figure 3.12) is obtained without knowing the exact parameters for the CS Saucer, it gives a good indication on the coupling effects. As seen, there is a clear pitching when surge force is applied. This effect might be small, but due to the strong coupling with the pendulum, as shall be shown in the continuing, it is included in the model.

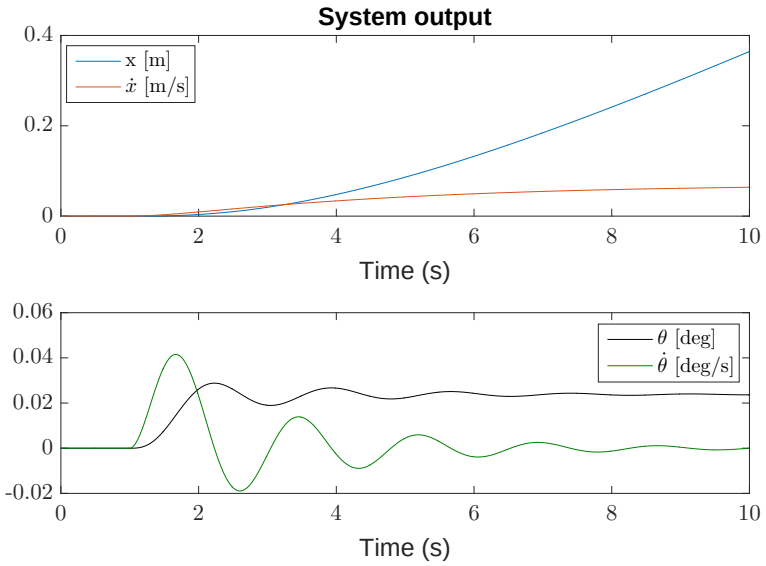


Figure 3.12: x position and surge velocity (top); pitch angle and rate (bottom) when the CS Saucer is subjected to a step in commanded surge force.

The parameters used in the simulations are given in Appendix C.

3.4 Marine Inverted Pendulum

While Sharoni (2015) used Newton's laws to derive (3.4), the Lagrangian approach will be used to derive the equations for the marine inverted pendulum. Knowing that the result should be similar, some of the equations have been confirmed with Newton's laws.

The Lagrange equation is

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = Q \ , \tag{3.16}$$

where q_j is the j-th variable, $L = T - P$ is the Lagrangian, which is the potential energy P subtracted from the kinetic T. Q is a vector of generalized forces or moments, not accounted for in the energy expressions.

It is important to mention that (3.16) is valid in any reference frame, as long as q_j is a generalized coordinate (Fossen, 2011, Ch. 6.3). The inertial frame, {n} is used in this thesis. Notice that the forces and moments Q are normally given in the {b} frame. Hence, a transformation must be applied to express these in {n}. A detailed discussion of the Lagrange mechanism can be found in Egeland and Gravdahl (2002, Ch. 8).

In the following the equations will be derived, showing that the forces and moments on the system can be written as a superposition:

$$\tau = \tau_{\text{MIP}} + \tau_{\text{Hydro}} + \tau_{\text{Thrusters}} \ . \tag{3.17}$$

3.4.1 Rigid Body Kinematics

Before (3.16) can be applied, the kinematic relation for the MIP must be clear. The relevant parameters and variables are defined in Figure 3.13. Notice also that a new reference frame {p} is introduced at the pendulum joint.

While the frames in Figure 3.13 are according to the marine convention, inverted pendulum equations are usually derived in a coordinate system as depicted in Figure 3.14. Hence, the equations will be derived using the definitions of Figure 3.14.

Below, the equations for the MIP are derived using geometry. In Section 3.4.3 an alternative representation, inspired from robotics, is presented.

The system is divided into two rigid body elements:

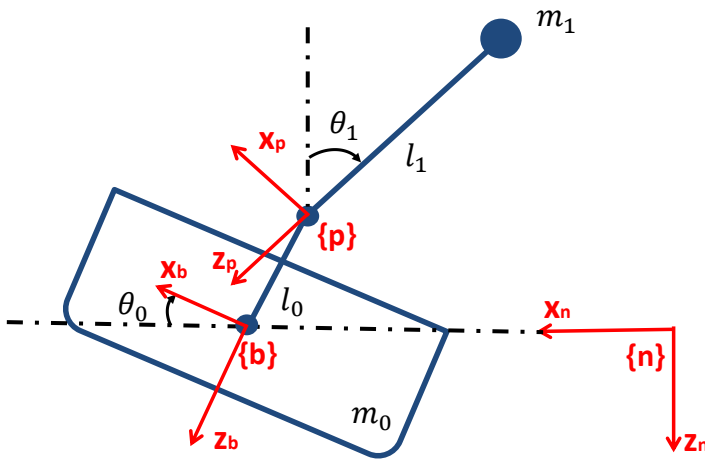


Figure 3.13: Marine inverted pendulum. Definition of symbols and variables, and relevant reference frames.

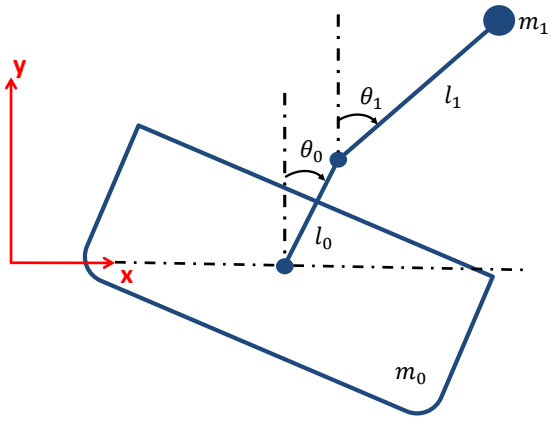


Figure 3.14: Frame and notation used in the derivation of the MIP equations.

- The vessel, including the fixed pendulum stand, with mass m_0, inertia I_0, and arm l_0 from centre of mass/rotation of the vessel to pendulum hinge.
- The pendulum, with mass m_1, inertia I_1, and arm l_1.

The considered degrees of freedom are then:

- Vessel surge translation, x_0, relative the origin.
- Vessel pitch rotation, θ_0, relative to the upright horizontal.
- Pendulum joint rotation, θ_1, relative to the upright horizontal.

The assumption is used that the centre of mass coincides with the centre of rotation for the vessel.

This then defines a state vector

$$x = \begin{bmatrix} x_0 \\ \theta_0 \\ \theta_1 \end{bmatrix} . \tag{3.18}$$

The vessel heave translation, y_0 is assumed fixed.

From this, the position of the centre of mass of the vessel is

$$p_0 = \begin{bmatrix} x_0 \\ 0 \end{bmatrix} . \tag{3.19}$$

Consequently, the vessel velocity is

$$v_0 = \frac{d}{dt} p_0 = \begin{bmatrix} \dot{x}_0 \\ 0 \end{bmatrix} . \tag{3.20}$$

The pendulum centre of mass position is

$$p_1 = \begin{bmatrix} x_0 + l_0 \sin\theta_0 + l_1 \sin\theta_1 \\ 0 + l_0 \cos\theta_0 + l_1 \cos\theta_1 \end{bmatrix} , \tag{3.21}$$

and consequently, the pendulum velocity is

$$v_1 = \frac{d}{dt} p_1 = \begin{bmatrix} \dot{x}_0 + l_0 \cos\theta_0 \dot{\theta}_0 + l_1 \cos\theta_1 \dot{\theta}_1 \\ 0 - l_0 \sin\theta_0 \dot{\theta}_0 - l_1 \sin\theta_1 \dot{\theta}_1 \end{bmatrix} . \tag{3.22}$$

3.4.2 Rigid Body Kinetics

The vessel kinetic energy is translation and rotational,

$$\begin{aligned} T_0 &= \frac{1}{2} m_0 \left\| v_0 \right\|^2 + \frac{1}{2} I_0 \dot{\theta}_0^2 \\ &= \frac{1}{2} m_0 \dot{x}_0^2 + \frac{1}{2} I_0 \dot{\theta}_0^2 \ , \end{aligned} \tag{3.23}$$

while its potential energy is

$$P_0 = m_0 g y_0 = 0 \tag{3.24}$$

when only considering the effect due to gravity. Potential restoring energy due to hydrodynamics will be included through generalized forces.

As for the vessel, the pendulum kinetic energy is translational and rotational,

$$\begin{aligned} T_1 &= \frac{1}{2} m_1 \left\| v_1 \right\|^2 + \frac{1}{2} I_1 \dot{\theta}_1^2 \\ &= \frac{1}{2} m_1 \left(\left(\dot{x}_0 + l_0 \cos\theta_0 \dot{\theta}_0 + l_1 \cos\theta_1 \dot{\theta}_1 \right)^2 + \left(l_0 \sin\theta_0 \dot{\theta}_0 + l_1 \sin\theta_1 \dot{\theta}_1 \right)^2 \right) \\ &\quad + \frac{1}{2} I_1 \dot{\theta}_1^2 \\ &= \frac{1}{2} m_1 \left(\dot{x}_0^2 + 2\dot{x}_0 l_0 \cos\theta_0 \dot{\theta}_0 + 2\dot{x}_0 l_1 \cos\theta_1 \dot{\theta}_1 + 2 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 + l_0^2 \dot{\theta}_0^2 + l_1^2 \dot{\theta}_1^2 \right) \\ &\quad + \frac{1}{2} I_1 \dot{\theta}_1^2 \\ &= \frac{1}{2} m_1 \left(\dot{x}_0^2 + 2\dot{x}_0 l_0 \cos\theta_0 \dot{\theta}_0 + 2\dot{x}_0 l_1 \cos\theta_1 \dot{\theta}_1 + 2 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 + l_0^2 \dot{\theta}_0^2 \right) \\ &\quad + \frac{1}{2} \left(m_1 l_1^2 + I_1 \right) \dot{\theta}_1^2 \end{aligned} \tag{3.25}$$

and its potential energy is

$$P_1 = m_1 g y_1 = m_1 g \left(l_0 \cos\theta_0 + l_1 \cos\theta_1 \right), \tag{3.26}$$

due to gravity. Hydrodynamics are included through generalized forces.

Now, the Lagrangian can be found as

$$\begin{aligned}
L &= T_0 + T_1 - P_0 - P_1 \\
&= \frac{1}{2} m_0 \dot{x}_0^2 + \frac{1}{2} I_0 \dot{\theta}_0^2 \\
&\quad + \frac{1}{2} m_1 \left({\dot{x}_0}^2 + 2\dot{x}_0 l_0 \cos\theta_0 \dot{\theta}_0 + 2\dot{x}_0 l_1 \cos\theta_1 \dot{\theta}_1 + 2 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 + l_0^2 \dot{\theta}_0^2 \right) \\
&\quad + \frac{1}{2} \left(m_1 l_1^2 + I_1 \right) \dot{\theta}_1^2 - m_1 g \left(l_0 \cos\theta_0 + l_1 \cos\theta_1 \right) \\
&= \frac{1}{2} \left(m_0 + m_1 \right) \dot{x}_0^2 + \frac{1}{2} \left(I_0 + m_1 l_0^2 \right) \dot{\theta}_0^2 + \frac{1}{2} \left(m_1 l_1^2 + I_1 \right) \dot{\theta}_1^2 \\
&\quad + m_1 l_0 \cos\theta_0 \dot{\theta}_0 \dot{x}_0 + m_1 l_1 \cos\theta_1 \dot{\theta}_1 \dot{x}_0 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 \\
&\qquad - m_1 g \left(l_0 \cos\theta_0 + l_1 \cos\theta_1 \right) \ .
\end{aligned} \tag{3.27}$$

Then, evaluating (3.16) for each of the three variables:

$$\begin{aligned}
X &= \frac{d}{dt} \left(\frac{\partial}{\partial \dot{x}_0} L \right) - \frac{\partial}{\partial x_0} L \\
&= \frac{d}{dt} \left(\left(m_0 + m_1 \right) \dot{x}_0 + m_1 l_0 \cos\theta_0 \dot{\theta}_0 + m_1 l_1 \cos\theta_1 \dot{\theta}_1 \right) - 0 \\
&= \left(m_0 + m_1 \right) \ddot{x}_0 - m_1 l_0 \sin\theta_0 \dot{\theta}_0^2 + m_1 l_0 \cos\theta_0 \ddot{\theta}_0 - m_1 l_1 \sin\theta_1 \dot{\theta}_1^2 \\
&\qquad + m_1 l_1 \cos\theta_1 \ddot{\theta}_1 \\
&= \left(m_0 + m_1 \right) \ddot{x}_0 + m_1 l_0 \cos\theta_0 \ddot{\theta}_0 + m_1 l_1 \cos\theta_1 \ddot{\theta}_1 - m_1 l_0 \sin\theta_0 \dot{\theta}_0^2 \\
&\qquad - m_1 l_1 \sin\theta_1 \dot{\theta}_1^2 \ ,
\end{aligned} \tag{3.28}$$

$$\begin{aligned}
M &= \frac{d}{dt} \left(\frac{\partial}{\partial \dot{\theta}_0} L \right) - \frac{\partial}{\partial \theta_0} L \\
&= \frac{d}{dt} \left(\left(I_0 + m_1 l_0^2 \right) \dot{\theta}_0 + m_1 l_0 \cos\theta_0 \dot{x}_0 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \dot{\theta}_1 \right) \\
&\qquad - \left(-m_1 l_0 \sin\theta_0 \dot{\theta}_0 \dot{x}_0 - m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 + m_1 g l_0 \sin\theta_0 \right) \\
&= \left(I_0 + m_1 l_0^2 \right) \ddot{\theta}_0 - m_1 l_0 \sin\theta_0 \dot{\theta}_0 \dot{x}_0 + m_1 l_0 \cos\theta_0 \ddot{x}_0 \\
&\qquad - m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right) \left(\dot{\theta}_0 - \dot{\theta}_1 \right) \dot{\theta}_1 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \ddot{\theta}_1 \\
&\qquad + m_1 l_0 \sin\theta_0 \dot{\theta}_0 \dot{x}_0 + m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right) \dot{\theta}_0 \dot{\theta}_1 - m_1 g l_0 \sin\theta_0 \\
&= m_1 l_0 \cos\theta_0 \ddot{x}_0 + \left(I_0 + m_1 l_0^2 \right) \ddot{\theta}_0 \\
&\qquad + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \ddot{\theta}_1 + m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right) \dot{\theta}_1^2 - m_1 g l_0 \sin\theta_0 \ ,
\end{aligned} \tag{3.29}$$

and

$$\begin{aligned}
0 &= \frac{d}{dt}\left(\frac{\partial}{\partial\dot{\theta}_1}L\right) - \frac{\partial}{\partial\theta_1}L \\
&= \frac{d}{dt}\left(\left(m_1 l_1^2 + I_1\right)\dot{\theta}_1 + m_1 l_1 \cos\theta_1 \dot{x}_0 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right)\dot{\theta}_0\right) \\
&\quad - \left(-m_1 l_1 \sin\theta_1 \dot{\theta}_1 \dot{x}_0 + m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\dot{\theta}_0\dot{\theta}_1 + m_1 g l_1 \sin\theta_1\right) \\
&= \left(m_1 l_1^2 + I_1\right)\ddot{\theta}_1 - m_1 l_1 \sin\theta_1 \dot{\theta}_1 \dot{x}_0 + m_1 l_1 \cos\theta_1 \ddot{x}_0 \\
&\quad - m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\left(\dot{\theta}_0 - \dot{\theta}_1\right)\dot{\theta}_0 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right)\ddot{\theta}_0 \\
&\quad + m_1 l_1 \sin\theta_1 \dot{\theta}_1 \dot{x}_0 - m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\dot{\theta}_0\dot{\theta}_1 - m_1 g l_1 \sin\theta_1 \\
&= m_1 l_1 \cos\theta_1 \ddot{x}_0 + m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right)\ddot{\theta}_0 + \left(m_1 l_1^2 + I_1\right)\ddot{\theta}_1 \\
&\quad - m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\dot{\theta}_0^2 - m_1 g l_1 \sin\theta_1 \ .
\end{aligned} \tag{3.30}$$

Equations (3.28), (3.29) and (3.30) can be written on matrix form as

$$\begin{aligned}
\underbrace{\begin{bmatrix} X \\ M \\ 0 \end{bmatrix}}_{\tau_{\mathrm{MIP}}} = {} & \underbrace{\begin{bmatrix} m_0 + m_1 & m_1 l_0 \cos\theta_0 & m_1 l_1 \cos\theta_1 \\ m_1 l_0 \cos\theta_0 & I_0 + m_1 l_0^2 & m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) \\ m_1 l_1 \cos\theta_1 & m_1 l_0 l_1 \cos\left(\theta_0 - \theta_1\right) & m_1 l_1^2 + I_1 \end{bmatrix}}_{D(x)} \underbrace{\begin{bmatrix} \ddot{x}_0 \\ \ddot{\theta}_0 \\ \ddot{\theta}_1 \end{bmatrix}}_{\ddot{x}} \\
& + \underbrace{\begin{bmatrix} 0 & -m_1 l_0 \sin\theta_0 \dot{\theta}_0 & -m_1 l_1 \sin\theta_1 \dot{\theta}_1 \\ 0 & 0 & m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\dot{\theta}_1 \\ 0 & -m_1 l_0 l_1 \sin\left(\theta_0 - \theta_1\right)\dot{\theta}_0 & 0 \end{bmatrix}}_{C(x,\dot{x})} \underbrace{\begin{bmatrix} \dot{x}_0 \\ \dot{\theta}_0 \\ \dot{\theta}_1 \end{bmatrix}}_{\dot{x}} \\
& + \underbrace{\begin{bmatrix} 0 \\ -m_1 g l_0 \sin\theta_0 \\ -m_1 g l_1 \sin\theta_1 \end{bmatrix}}_{G(x)} ,
\end{aligned} \tag{3.31}$$

or, not expanding the matrices,

$$\tau_{\mathrm{MIP}} = D\left(x\right)\ddot{x} + C\left(x, \dot{x}\right)\dot{x} + G\left(x\right) \ . \tag{3.32}$$

Assuming $D(x)$ is invertible, this gives the differential equation

$$\ddot{x} = D^{-1}\left(x\right)\left(-C\left(x, \dot{x}\right)\dot{x} - G\left(x\right) + \tau_{\mathrm{MIP}}\right) \ . \tag{3.33}$$

In fact, $D(x)$ is a physical matrix, that is all the elements have a physical interpretation. That means that all the elements in the matrix are positive values, within

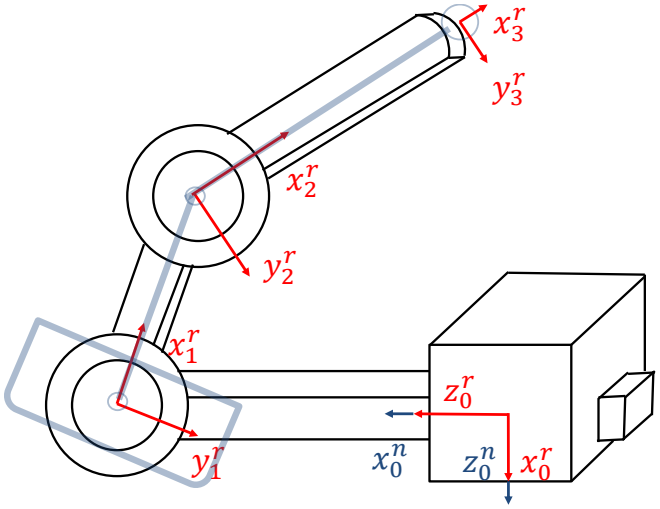


Figure 3.15: Interpretation of the MIP system as an industrial robot with one prismatic joint (from {n} to {b}), and two rotational joints.

the physical limits for θ_0 and θ_1. So the determinant can not be zero, and hence the inverse should indeed exist.

Moreover, (3.32) has the same form as the equations for the double inverted pendulum described by Bogdanov (2004, Eqn. 2), Zare et al. (2009, Eqn. 2) and Hassanzadeh et al. (2011, Eqn. 2.1). The MIP system indeed resembles a double inverted pendulum, θ_0 being the angle of the first pendulum, and θ_1 of the second. Hence, the results presented here are strengthened by the theory developed for the double inverted pendulum.

3.4.3 Alternative Representation

The system depicted in Figure 3.13 can be interpreted as an industrial manipulator with one prismatic joint, and two rotational joints. The joint variables will then be x_0, θ_0 and θ_1. Reference is made to Spong et al. (2006) for a detailed discussion on robotics, explaining the expressions used.

First of all, in robotics there exists a well established norm for deriving the system transformation matrices. This is referred to as the Denavit-Hartenberg (DH) convention (Spong et al., 2006, Ch. 1), where certain rules for placing and orientating the different coordinate frames apply. This is not in accordance with the marine convention. However, as seen from Figure 3.15, to get from the "robot" reference frame $(x_0^r,\ y_0^r,\ z_0^r)$ to {n}, a simple rotation must be applied.

Table 3.2: Table of parameter values for all the joints following the Denavit-Hartenberg conventions.

Joint	θ	a	d	α
1	180°	0	d_1*	-90°
2	θ_0*	l_0	0	0
3	θ_1*	l_1	0	0

*Joint variable.

The Denavit-Hartenberg parameters for the industrial manipulator depicted in Figure 3.15 are listed in Table 3.2.

Using Table 3.2 the homogeneous transformation matrices are found according to Spong et al. (2006, Ch. 3.2) as

$$T_1^0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.34}$$

$$T_2^1 = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 & 0 & l_0\cos\theta_0 \\ \sin\theta_0 & \cos\theta_0 & 0 & l_0\sin\theta_0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.35}$$

and

$$T_3^2 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & l_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & l_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.36}$$

The final transformation matrix, expressing the location (and orientation) of the tip of the pendulum is then

$$\begin{aligned} T_3^0 &= T_1^0 T_2^1 T_3^2 \\ &= \begin{bmatrix} -\cos(\theta_0+\theta_1) & \sin(\theta_0+\theta_1) & 0 & -l_1\cos(\theta_0+\theta_1) - l_0\cos\theta_0 \\ 0 & 0 & -1 & 0 \\ -\sin(\theta_0+\theta_1) & -cos(\theta_0+\theta_1) & 0 & d_1 - l_0\sin\theta_0 - l_1\sin(\theta_0+\theta_1) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned} \tag{3.37}$$

where the final result is obtained by applying properties for the trigonometric functions found in Rottmann (2010).

Notice that (3.37) is given in the {r} frame. To express the location of the pendulum tip in {n}, a final transformation must be applied, giving

$$
\begin{aligned}
T^n_{3/0} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T^0_3 \\
&= \begin{bmatrix} -\sin(\theta_0+\theta_1) & -\cos(\theta_0+\theta_1) & 0 & d_1 - l_0\sin\theta_0 - l_1\sin(\theta_0+\theta_1) \\ 0 & 0 & -1 & 0 \\ -\cos(\theta_0+\theta_1) & \sin(\theta_0+\theta_1) & 0 & -l_1\cos(\theta_0+\theta_1) - l_0\cos\theta_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.38}
$$

The first and third entries in the last column in (3.38) are the location of the pendulum tip, expressed in {n}. By comparison, these terms are equal to (3.21), with the transformation of coordinate frames used here.

Now, Egeland and Gravdahl (2002, Ch. 8), and especially Spong et al. (2006, Ch. 7) shows how the Lagrange equation can be derived for a robotic manipulator as depicted in Figure 3.15. Under the assumptions that the kinetic energy is a quadratic function of the joint variables (x in (3.32)), and that the potential energy is independent of $\dot{x}$, Spong et al. (2006, Ch. 7) shows that the Euler-Lagrange equation can be written in the exact form of (3.32).

Hence, this section shows how an untraditional approach from another field of study can be applied for the inverted pendulum. Although not further exploited in this thesis, this supports the theory covered in Section 3.4.1 and 3.4.2.

3.4.4 Hydrodynamics

Towards the end of Section 3.3 it was mentioned that (3.10) must be expressed in {n} before it could be combined with (3.31). However, for the 2 DOF model, the two equations can be combined directly. This can be verified by setting up the mass-damper-spring system representing the vessel, according to Newton's second law in the Figure 3.14 frame.

So, under the aforementioned assumption, that the vessel is performing strictly in plane motions, (3.10) can be rewritten directly to {n} in a compact form as:

$$
M_{\mathrm{RB}}\ddot{x} = -D(\dot{x})\dot{x} - C(\dot{x})\dot{x} - g(x) - M_{\mathrm{A}}\ddot{x} + \tau_{\text{thruster}} \ , \tag{3.39}
$$

where $x = [x_0 \ \theta_0]^T$ in this specific equation.

Now, the vector τ in (3.32) contains all external forces not captured in the kinetic and potential energy derivations earlier in this section. These are all the hydrodynamic forces, as well as the thruster forces, that is, the right hand side of (3.39). Consequently, by Newton's second law. the following must be true under the stated assumptions:

$$\tau_{\text{MIP}} = M_{\text{RB}}\ddot{x} \ , \tag{3.40}$$

Where M_{RB} is expanded from (3.6) to the 3 by 3 matrix

$$M_{\text{A}} = \begin{bmatrix} M_{\text{RB}} & 0_{2\times 1} \\ 0_{1\times 2} & 0 \end{bmatrix} \ . \tag{3.41}$$

Expanding the other hydrodynamic matrices from (3.39) in a similar fashion, while adding a subscript H to the hydrodynamic matrices, and P to the pendulum matrices (3.32), yields the following combined system:

$$D_{\text{P}}(x)\ddot{x} = \tau_{\text{thruster}} - \underbrace{[M_{\text{AH}}\ddot{x} + D_{\text{H}}\dot{x} + C_{\text{H}}\dot{x} + g_{\text{H}}(x)]}_{\text{Hydrodynamic forces}} - \underbrace{[C_{\text{P}}(x,\dot{x})\dot{x} + G_{\text{P}}(x)]}_{\text{Pendulum dynamic forces}} \ . \tag{3.42a}$$

Rearranging the therms to find the system matrices gives

$$\tau_{\text{thruster}} = \underbrace{[D_{\text{P}}(x) + M_{\text{AH}}]}_{D(x)}\ddot{x} + \underbrace{[C_{\text{P}}(x,\dot{x}) + D_{\text{H}}(\dot{x}) + C_{\text{H}}(\dot{x})]}_{C(x,\dot{x})}\dot{x} + \underbrace{[G_{\text{P}}(x) + g_{\text{H}}(x)]}_{G(x)} \ , \tag{3.42b}$$

with the matrices

$$D(x) = \begin{bmatrix} m_{11\text{A}} + m_0 + m_1 & m_{15\text{A}} + m_1 l_0 \cos\theta_0 & m_1 l_1 \cos\theta_1 \\ m_{51\text{A}} + m_1 l_0 \cos\theta_0 & m_{55\text{A}} + I_0 + m_1 l_0^2 & m_1 l_0 l_1 \cos(\theta_0 - \theta_1) \\ m_1 l_1 \cos\theta_1 & m_1 l_0 l_1 \cos(\theta_0 - \theta_1) & m_1 l_1^2 + I_1 \end{bmatrix} \ , \tag{3.43a}$$

$$C(x,\dot{x}) = \begin{bmatrix} c_{11} + d_{11} & c_{15} + d_{15} - m_1 l_0 \sin\theta_0 \dot{\theta}_0 & -m_1 l_1 \sin\theta_1 \dot{\theta}_1 \\ c_{51} + d_{51} & c_{55} + d_{55} & m_1 l_0 l_1 \sin(\theta_0 - \theta_1)\dot{\theta}_1 \\ 0 & -m_1 l_0 l_1 \sin(\theta_0 - \theta_1)\dot{\theta}_0 & 0 \end{bmatrix} \ , \tag{3.43b}$$

and

$$G(x) = \begin{bmatrix} g_{11} \\ (g_{55} - m_1 g l_0)\sin\theta_0 \\ -m_1 g l_1 \sin\theta_1 \end{bmatrix} \ . \tag{3.43c}$$

Observe that (3.42a) is on the form presented in (3.17).

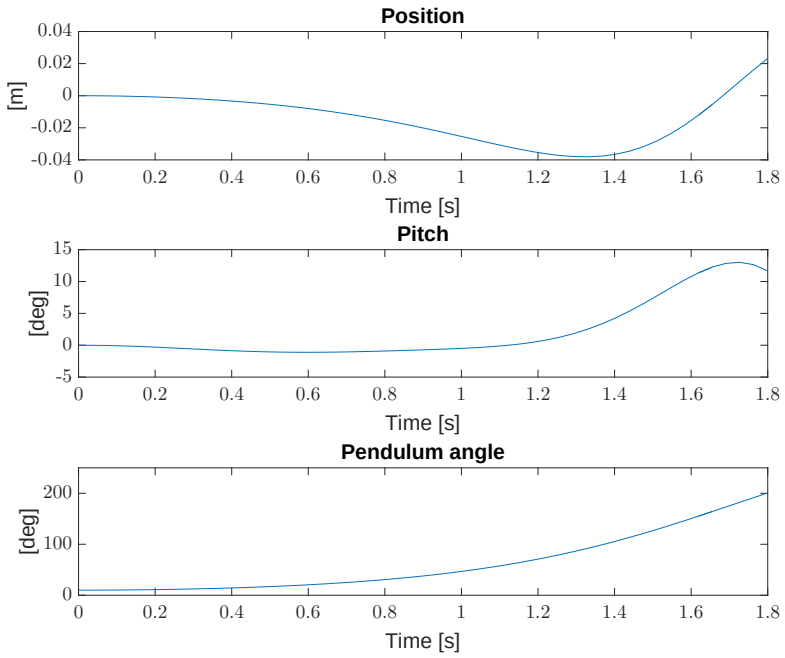


Figure 3.16: Pendulum subsystem simulation. Pendulum initialized at 10 degrees, and zero position and pitch.

3.4.5 Marine Inverted Pendulum Simulation Model

To verify the correctness of the MIP equations, a model has piecewise been built in Simulink and MATLAB. First, (3.31) has been implemented, without any hydrodynamics, as attached in Appendix A.6 and A.7. Notice that a restoring force in pitch is added to avoid having the model pitching round and round.

Figure 3.16 shows the behaviour of the system. Notice that this is given in the framework depicted in Figure 3.14. The pendulum is initiated at $\theta_1 = 10°$, while the position and pitch angle are both set to zero. Intuitively one would expect the pendulum to fall over until it reaches its limit. Due to the force balance at the pendulum hinge, this will force the vessel to move in a negative x-direction, as also verified in Section 3.2.

Figure 3.16 clearly shows that the pendulum falls over. This causes, as expected, the vessel to go in a negative direction. The force from the pendulum falling over is translated to the vessel at the pendulum hinge point, resulting in a negative pitching moment (according to the notation of Figure 3.14). This is seen as a negative

pitch angle initially. After the pendulum has reached the limit, it causes the vessel to pitch in a positive direction and bear of with a positive x displacement. This is also expected as there is no hydrodynamic or external forces acting now. That is, there is no damping or other forces from the water stopping the movement.

Introducing the physical range limiter for the pendulum motion gives discrete dynamics that has to be properly analysed, and is out of the scope of this thesis. Hence, the pendulum is allowed to fall freely. This means that it is only worth examining the first few seconds of the uncontrolled system response. The control objective is to keep the pendulum well within the valid limit, so the model is sufficient for controller design.

Now, for the complete system (3.42) where all the dynamics are included, a simulation model and run file are attached in Appendix A.8 and A.9.

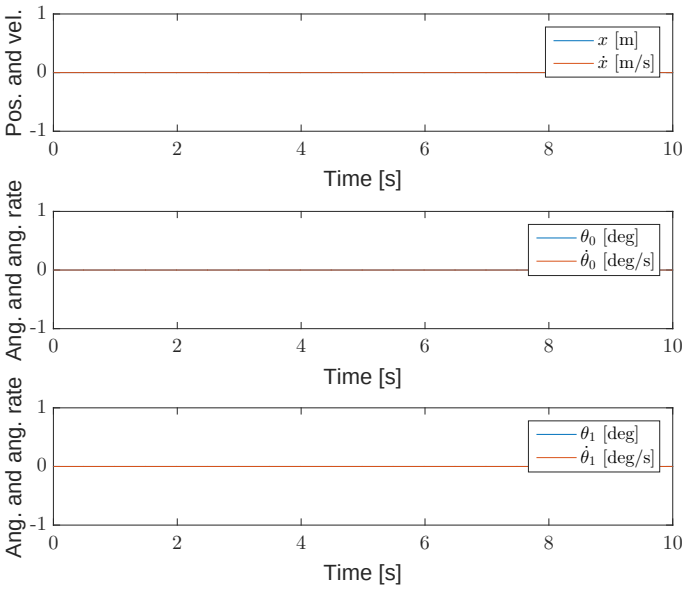
So, under all the aforementioned assumptions, three distinct cases are shown here to validate the theory developed in the whole Section 3.4:

1. All states are initialized to zero, and there is no forces from the thrusters.
2. All three states of the system are initialized to zero, and the vessel is given a constant force in surge (and hence also moment in pitch).
3. The thruster forces are zero, but the pendulum is initialized with an angle different from zero.

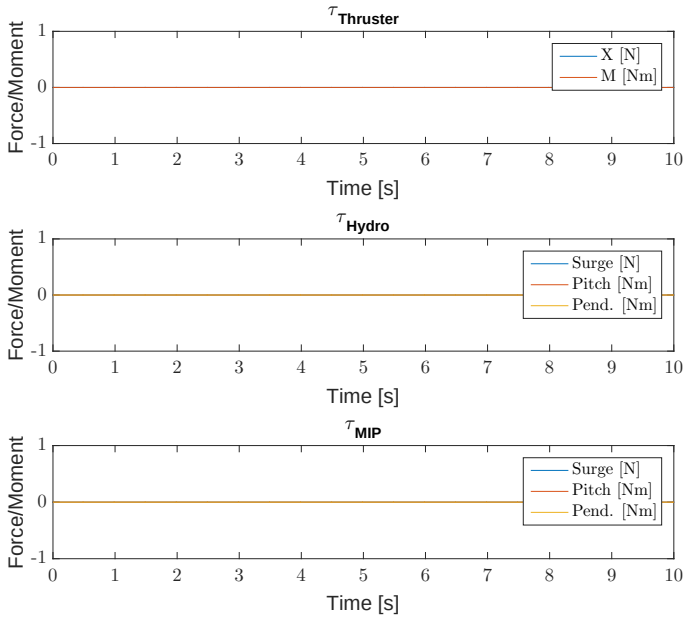
The first case is presented in Figure 3.17, and shows that the system indeed remains at rest. This indicates that the upright equilibrium pendulum indeed is a stable position for the system.

Case 2 is presented in Figure 3.18. Figure 3.18b shows each force component as grouped in (3.42a). This verifies that the hydrodynamic forces are incorporated into the complete system with right sign. They act in the same direction as the thruster forces, but are subtracted in the equation. Moreover they are smaller in size. This is well in accordance with theory and shown practice from hydrodynamics.

Now over to Figure 3.18a showing the states and derivatives for the system. The upper plot shows the position and speed increasing. This will continue until hydrodynamic forces are equal to the thruster forces, when the vessel will continue with constant speed. Due to the moment from the thruster, one can also observe in the second plot that the vessel is pitching with a positive pitch angle. At the same time, the pendulum is falling over in the opposite direction of the vessel motion. This is then all as could be expected, hence strengthening the theoretical derivations.

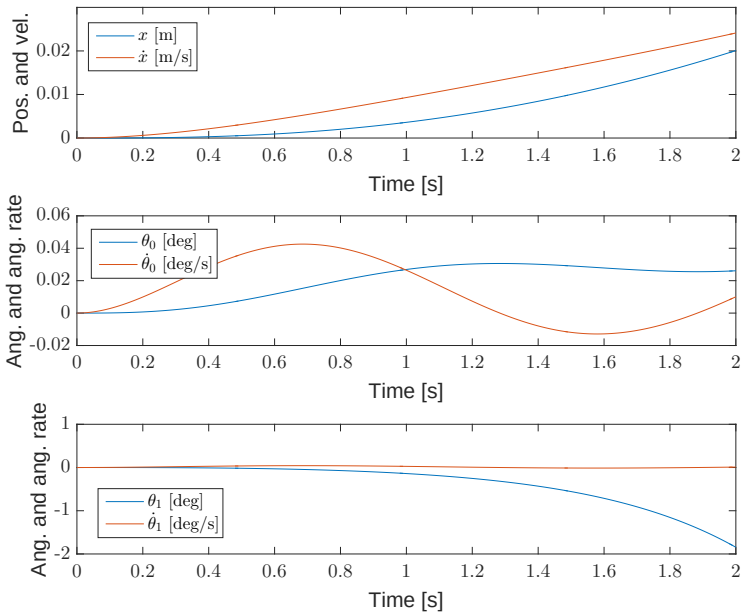


(a) States of the MIP with hydrodynamic forces, and the corresponding velocities.

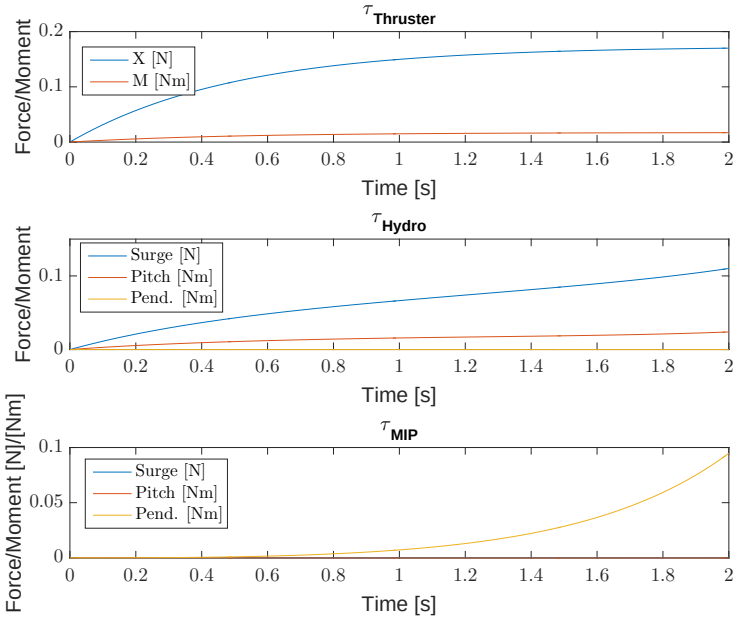


(b) Forces and moments on the system as divided in (3.42a).

Figure 3.17: Case 1: All states initialized at zero, and no thruster actuation.



(a) States of the MIP with hydrodynamic forces, and the corresponding velocities.

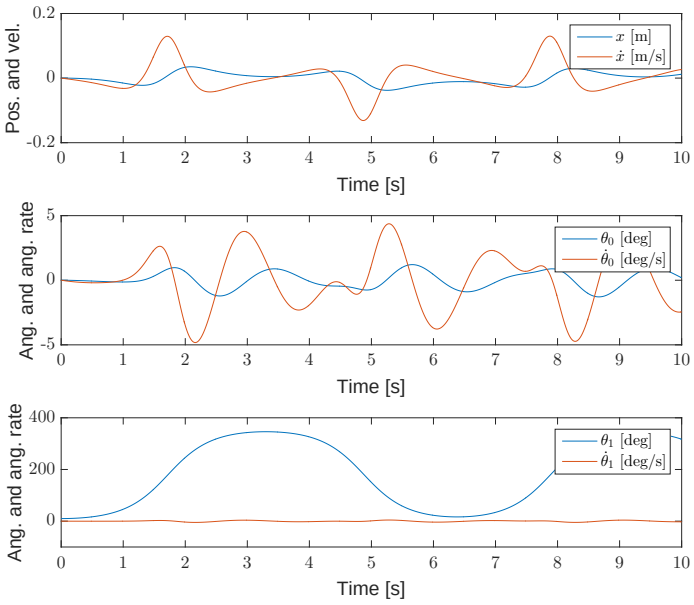


(b) Forces and moments on the system as divided in (3.42a).

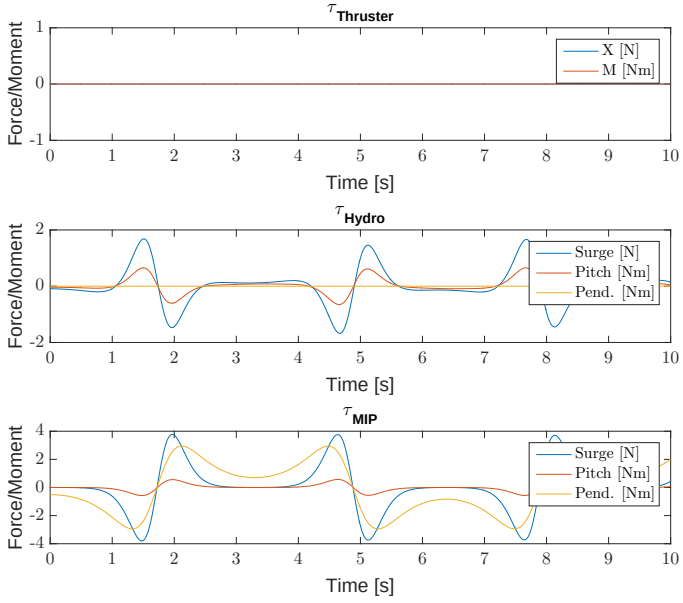
Figure 3.18: Case 2: Constant thruster force and all states initialized to zero.

Finally, case 3 is presented in Figure 3.19, when the pendulum is initialized at 10°, and no thruster force is applied. Since there is no physical limitation added to the model, the pendulum swings back and forth trough 360°. This causes the vessel to pitch both ways, and the position to oscillate slightly.

As earlier, not all the exact parameters are known. They are therefore estimated or taken as given by Idland (2015). Moreover, in general, the damping matrix is hard to find, so the approximation made by Fossen (2011, Ch. 7.5) that the coupling terms are small in general (and hence neglected) is used here. Parameters used can be found in Appendix C.



(a) States of the MIP with hydrodynamic forces, and the corresponding velocities.



(b) Forces and moments on the system as divided in (3.42a).

Figure 3.19: Case 3: Pendulum initialized at 10°. All other states, and thruster forces are zero.

Chapter 4

Controller and Observer Design

4.1 Linearized State-Space Model

To design an observer and controller for the marine inverted pendulum, it is desirable to express the system on the state space form

$$\begin{aligned} \dot{z} &= Az + B\mu \\ y &= Cz \ , \end{aligned} \tag{4.1}$$

where $z = [x \ \dot{x}]^T \in \mathbb{R}^6$ is the state vector, consisting of the MIP states (3.18) and their derivatives. $A \in \mathbb{R}^{6\times 6}$ the state matrix and $B \in \mathbb{R}^6$ the input matrix. $\mu \in \mathbb{R}$ is the input to the system, that is the surge force from the thrusters. $y \in \mathbb{R}^m$ is the vector of measured states and $C \in \mathbb{R}^{m\times 6}$ is the measurement matrix, relating the true states to the ones that are measured.

Writing the system (3.42b) using $z \in \mathbb{R}^6$ as the state vector, omitting the dependencies of the states and their derivatives in the system matrices, yields

$$\begin{aligned} \dot{z} &= \begin{bmatrix} 0 & I \\ 0 & -D^{-1}C \end{bmatrix} z + \begin{bmatrix} 0 \\ -D^{-1}G \end{bmatrix} + \begin{bmatrix} 0 \\ D^{-1}H \end{bmatrix} \mu \\ y &= Cz \ , \end{aligned} \tag{4.2}$$

where the dimension of y and C will be discussed in Section 4.2. $H = [1 \ l_z \ 0]^T$ is a vector relating the thruster force to the states of the system. Such that the following is true:

$$\tau_{\text{thrusters}} = TK\mu = H\mu \tag{4.3}$$

Since the relation

$$\frac{\partial\left(-D^{-1}(x)G(x)\right)}{\partial x}x = -D^{-1}(x)G(x) \tag{4.4}$$

holds, a linearization about $z = 0$ gives the system on the state-space form (4.1), where the matrices A and B are given as

$$A = \begin{bmatrix} 0 & I \\ -D(0)^{-1}\frac{\partial G(0)}{\partial x} & 0 \end{bmatrix} \tag{4.5a}$$

$$B = \begin{bmatrix} 0 \\ D(0)^{-1}H \end{bmatrix}. \tag{4.5b}$$

The measurement matrix C is discussed in Section 4.2.

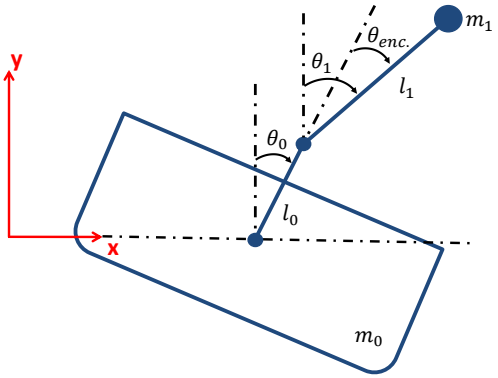


Figure 4.1: Pitch and pendulum angle, and angle measured by the encoder.

4.2 Observer Design

The measured values are position and pitch angle, given by the Qualisys system, and pendulum angle, given by the encoder. So, the measurement vector y will be three by one, and hence the matrix C in (4.2) will be a three by six matrix. Obviously, this could change at a later stage if other sensors are installed.

This means that only three states (out of six) are measured, but, for feedback control, as the LQR discussed in Section 4.3.2, all six are needed. Thus an observer is introduced to estimate, or reconstruct, unmeasured states. However, this is not the sole task of the observer. As the experiments are going to be conducted with a real system in the MC Lab, one would expect sensor noise and possibly sensor losses. The observer therefore also acts as a filter, filtering noise from the measurements. In the case of sensor loss it can also be used for dead reckoning, reconstructing lost states.

As mentioned, the encoder is rigidly attached to the vessel. Hence, for a pitching angle different from zero, the encoder will read a pendulum angle different from zero even tough the pendulum in fact is upright. So, in the experiments, the true pendulum angle is taken as

$$\theta_1 = \theta_0 + \theta_{\text{encoder}} \ , \tag{4.6}$$

since the measured pendulum angle will increase with decreasing pitch angle, as illustrated by Figure 4.1.

4.2.1 Luenberger Observer

In order to design a Luenberger Observer, the system must be observable. Observability means that the current state of the system can be reconstructed in finite time, using only the measured outputs and the known control input. A thorough study of observability was done by Sharoni (2015, Ch. 4.4).

So, given that the pair (A, C) are observable, a Luenberger observer can be created as

$$\dot{\hat{z}} = A\hat{z} + B\mu + L(y - C\hat{z}) \ . \tag{4.7}$$

Notice that this is a copy of the system dynamics, where system and measurement noise are neglected. The term $L(y - C\hat{z})$ is called an injection term where the gain L can be chosen by pole placement to achieve desired dynamics. In MATLAB this can be done by using the command `L = place(A',C',p)`, where `p` is a vector of desired poles.

4.2.2 Kalman Filter

The Kalman filter is an extension to the Luenberger observer, but the observer gain is updated online to give a minimum variance estimate. In combination with the LQR, the control and observer system with the Kalman filter solves the so called linear quadratic Gaussian (LQG) problem.

Since the Kalman Filter is challenging to implement in continuous-time, the discrete system will be described. The system is discretized using for instance zero-order hold. The discrete system matrices are denoted with subscript d. Once the discrete system is obtained, the Kalman Filter equations can be calculated as shown in the following.

First, the initial state estimate is expressed as $\hat{z}_k^-$, for $k = 0$. This is the initial condition $z(0)$. Furthermore the initial covariance matrix is given as

$$P_k^- = E[(z(0) - \hat{z}(0))(z(0) - \hat{z}(0))^T] \ , \ k = 0 \ , \tag{4.8}$$

where E denotes the expected value.

Then the Kalman filter gain is found as

$$L_k = P_k^- C_d^T (C_d P_k^- C_d^T + R)^{-1} \ , \tag{4.9}$$

and the state estimation and covariance matrix can be updated with the measurement:

$$\hat{z}_k = \hat{z}_k^- + L_k(y_k - C_d \hat{z}_k^-) \tag{4.10}$$

$$P_k = (I - L_k C_d) P_k^- (I - L_k C_d)^T + L_k R K_k^T \ . \tag{4.11}$$

Now, projecting ahead and calculating the new a priori state estimate and covariance matrix:

$$\hat{z}_{k+1}^- = A_d \hat{z}_k + B_d \mu_k \tag{4.12}$$

$$P_k^- = A_d P_k A_d^T + E_d Q E_d^T \ . \tag{4.13}$$

These equations are based on the linear system model (4.1), where noise is added, and the model is in discrete time:

$$\begin{aligned} z_{k+1} &= A_d z_k + B_d \mu_k + E_d w_k \\ y_k &= C_d z_k + v_k \ , \end{aligned} \tag{4.14}$$

where v and w are zero-mean white noise terms.

The matrices Q and R are called design matrices. The first is a measure of the process noise variance while the latter is the measurement noise variance. Both matrices must be positive definite. Notice that although the same names have been used, these matrices are not related to the Q and R presented in Section 4.3.2.

4.2.3 Extended Kalman Filter

The Kalman filter can also be extended to yield for nonlinear systems on the form

$$\begin{aligned} \dot{z} &= f(z) + B\mu + Ew \\ y &= Cz + v \ . \end{aligned} \tag{4.15}$$

The Kalman filter procedure is then as given in Section 4.2.2, after the nonlinear system have been discretized. The only difference being that (4.12) now is found as

$$\hat{z}_{k+1}^- = \mathcal{F}(\hat{z}_k, \mu_k) \ , \tag{4.16}$$

where $\mathcal{F}(z_k, \mu_k)$ is the discrete time version of $f\left(z(t), \mu(t)\right)$. This version is, not surprisingly, called the Extended Kalman filter.

4.3 Controller Design

The control objective is to bring the pendulum to the upright equilibrium, and to keep it there. Mathematically this can be expressed as

$$\text{Objective: } \lim_{t\to\infty} \theta_1(t) = 0 \ . \tag{4.17}$$

In this thesis the position and pitch angle are left uncontrolled. However, it follows naturally that they also tend to zero if the objective is met.

4.3.1 Heading Controller

Due to the shape of the CS Saucer, the damping in yaw is nearly naught. Furthermore, the thrust provided by the thrusters is slightly different due to the contra rotating rear propellers, mentioned in Section 3.3.1. This results in the CS Saucer being directionally unstable, which raises the need for a heading controller.

The sole aim of the heading controller is to maintain a stable and straight course when the vessel is commanded to pure surge motion. For this, a simple PD controller on the form

$$N_{\text{desired}} = -K_p(\psi - \psi_{\text{desired}}) - K_d r \tag{4.18}$$

is sufficient, where the relation $r = \dot{\psi}$ holds.

K_p and K_d are the controller gains. They are tuned to get the desired behaviour from the controller.

4.3.2 Linear Quadratic Regulator

The LQR apporach is based on minimizing the cost function

$$J = \int_0^\infty (z^T Q z + \mu^T R \mu + 2 z^T N \mu)\ dt \ , \tag{4.19}$$

where Q and $R, \in \mathbb{R}^{6\times 6}$, are positive definite matrices, normally chosen diagonal. So, by selecting the relation between Q and R, one can chose the relation between fast convergence to desired states, and required control input. This topic was covered more extensively by Sharoni (2015, Ch. 4.3), also including conditions for controllability.

So, if the pair (A, B) are controllable, a state feedback controller can be designed that, for any given initial condition, brings the system to a desired final state, in finite time. This is achieved by choosing the feedback gain as

$$\mu = -R^{-1}B^T P_c z := -K_c z \ , \tag{4.20}$$

where P_c is the steady state solution of the differential Ricatti Equation. In practice, this is the solution of the infinite-horizon algebraic Riccati Equation given as

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0 \ . \tag{4.21}$$

However, for practical implementations, the feedback gain K_c is found using the MATLAB call `[K,P,e] = lqr(A,B,Q,R,N)`. Using this approach, the weighing matrices Q and R must be chosen according to the desired response. The cross term, N is normally set to zero.

An important remark is that although the final state can be reached in finite time with this approach, the result is purely mathematical. In reality, this might require an unphysical control input μ.

4.3.3 State-Dependant Riccati Equation Control

The State-Dependant Riccati Equation (SDRE) is used on the pseudo linear form of the system equations, where the matrices are state dependant:

$$\dot{z} = A(z)z + B(z)\mu \ . \tag{4.22}$$

The same problem as for the LQR is solved, but at each time step, as opposed to the LQR. The feedback control law is then similar to the one given for the LQR (4.20), but the feedback gain is now state dependant, given as

$$\mu = -K(z_k)z_k \tag{4.23}$$

for the discrete time system. The notation z_k is the state at discrete time k: $z(t = k)$. Notice that the discrete state dependant matrices are treated as constants for each time step, and thus this approach can be seen as a linear extension to the LQR (Bogdanov, 2004). Moreover, the system (4.22) forms the linear equation (4.1) in the neighbourhood of the equilibrium, $z = 0$.

4.3.4 Feedback Linearization

Feedback linearization is basically transforming a nonlinear system to a linear one. This is the same as for the LQR controller, where the controller gain also can be decided by pole placement. However, the feedback linearization technique allows to transform the system to a very simple form, where $\ddot{x} = \mu$. This can be done by cancelling terms in the system equation by chosing the control input, μ.

Consider the MIP system with hydrodynamics, given by (3.42b):

$$\tau_{\text{thruster}} = D(x)\ddot{x} + C(x, \dot{x})\dot{x} + G(x) \ ,$$

all the nonlinearities can be cancelled by choosing the control law as

$$\tau_{\text{thruster}} = D(x)a_c + C(x, \dot{x})\dot{x} + G(x) \ , \tag{4.24}$$

where a_c is the commanded acceleration vector. This vector can then, according to Fossen (2011), be chosen by pole placement techniques, or for instance as a PID controller with acceleration feedforward on the form

$$a_c = \ddot{x}_d - K_d(\dot{x} - \dot{x}_d) - K_p(x - x_d) - K_i \int_0^t (x(\sigma) - x_d(\sigma))d\sigma \ , \tag{4.25}$$

where x_d is the desired state vector.

Notice that by selecting the control law (4.24), the system simplifies to $\ddot{x} = a_c$. Hence all the nonlinearities are cancelled in the controller.

4.3.5 Integrator Backstepping

Backstepping is a design technique that might be very applicable for the inverted pendulum system, as it allows to recursively design the controller. The controller can then be designed by starting with the isolated pendulum, then recursively work towards the vessel. The technique is very similar to feedback linearization, but backstepping allows to keep "good" nonlinearities in the controller (Fossen, 2011). "Good" nonlinearities can for instance bee the restoring term, as it helps keeping the pitching angle low. Moreover, feedback linearization requires good and accurate models of the system, which in practice are often hard to obtain.

The main idea behind backstepping is that one can start with designing a virtual controller, bringing the pendulum to the equilibrium from the hinge. The virtual control is designed such that it contains a stabilizing function, and a new state

variable. From this, a new virtual controller can be designed, bringing the pendulum hinge to a desired position, for instance. The final step will be to design the actual controller that controls the vessel position.

The method is based on Control Lyapunov Functions, and is well explained by Fossen (2011, Ch. 13.3). Therefore, integrator backstepping is mentioned here as it is believed to be a good algorithm for the MIP, however it will not be further exploited in the scope of this thesis.

Chapter 5

Results

5.1 Simulations

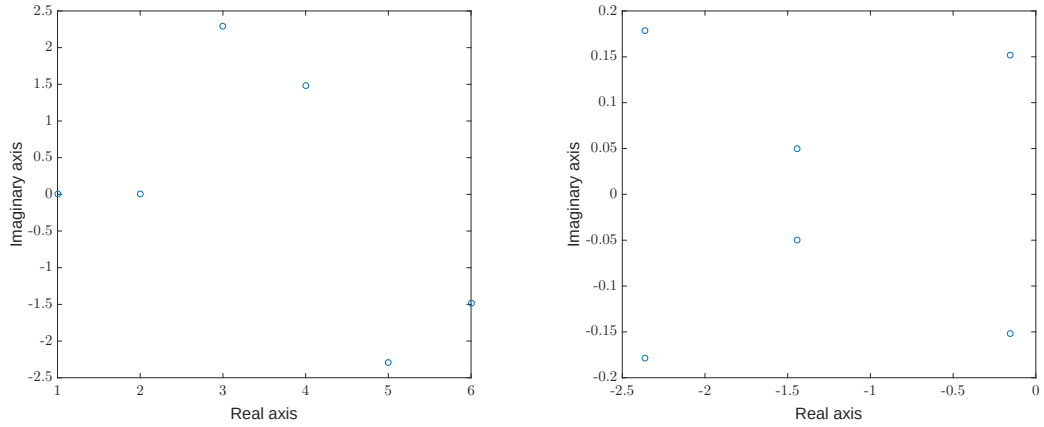
This section presents the results from simulated tests of the observer and controller, while the experimental results from the MC Lab follows in Section 5.2.

5.1.1 LQR

The LQR controller presented in Section 4.3.2 have been tested on (3.31) and on the complete system (3.42). For the first case, the open loop poles for the linear system are presented in Figure 5.1a. Those are the eigenvalues of A. Figure 5.1b shows the location of the closed loop poles in the complex plane, that is, the eigenvalues of $A - BK$.

Applying the LQR on (3.31), with desired state vector $z_d = 0$ yields the results in Figure 5.2.

Using the same controller on the system with hydrodynamic effects taken into account results in Figure 5.3. Also here the desired state is $z_d = 0$.



(a) Poles of the uncontrolled linear MIP system.

(b) Poles of the MIP system controlled with LQR.

Figure 5.1: Linear MIP system poles.

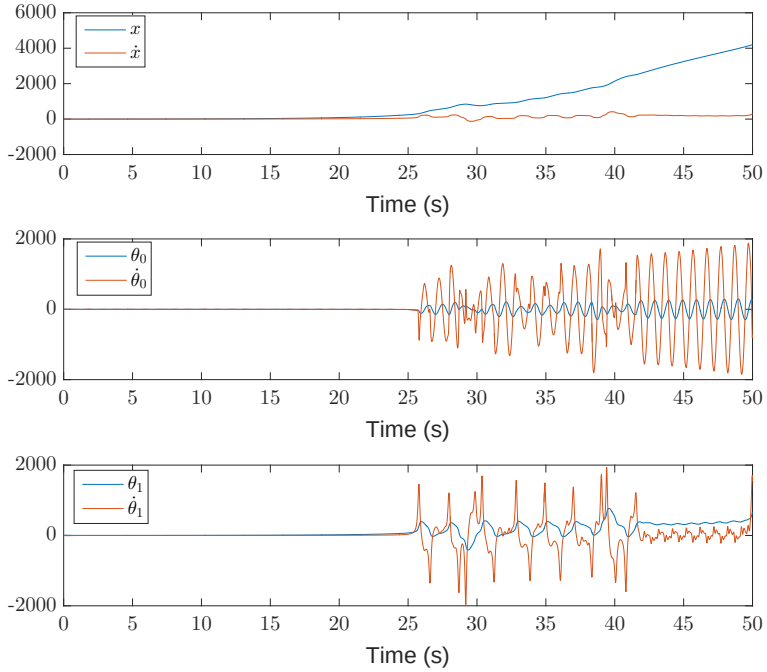


Figure 5.2: Simulation of the MIP system controlled with LQR.

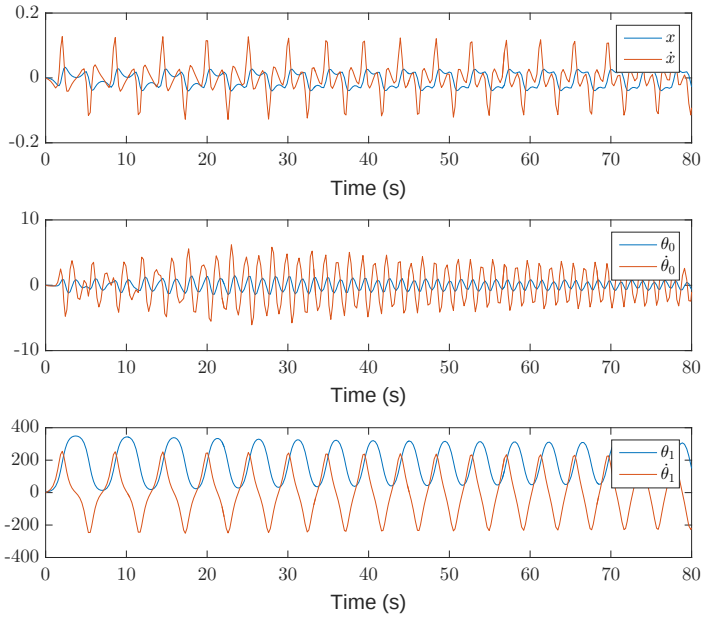


Figure 5.3: LQR applied to the MIP with hydrodynamic effects considered.

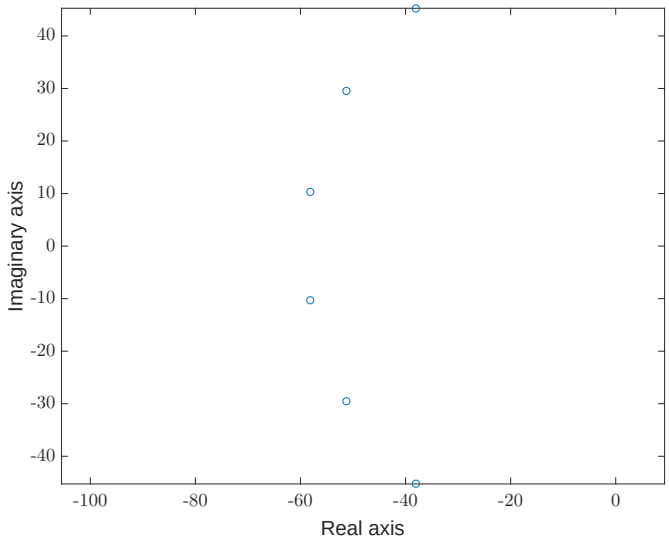


Figure 5.4: Observer poles along a half circle in the left half of the complex plane.

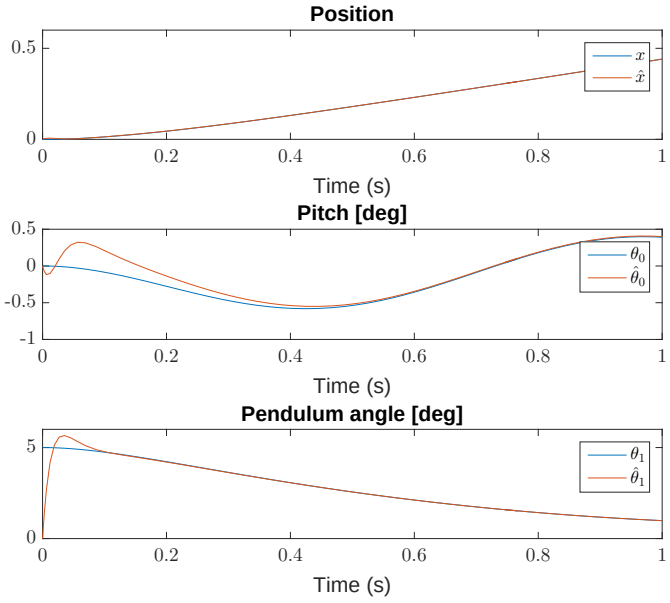
5.1.2 Luenberger Observer

The Luenberger observer is designed based on the MIP equations (3.31), where hydrodynamics are not considered. The poles are then placed on an half circle in the left complex plane. The location is shown in Figure 5.4.

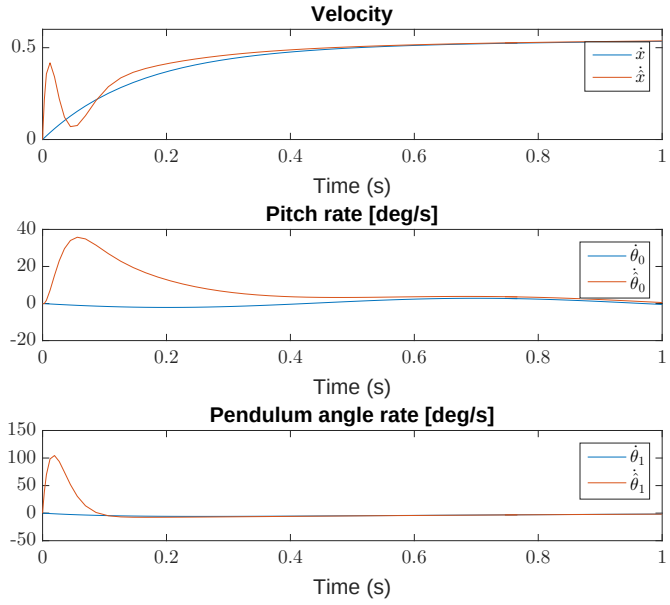
A simulation of the MIP is then run in Simulink, where the estimated states (in red) are compared to the real states (blue). The results are seen in Figure 5.5. The system is also controlled with an LQR in these results.

Although the graph is only showing the first second, a 20 second simulation shows that the estimated states are identical to the real ones. Only the first second is included to emphasize how the estimated states behave initially. Nevertheless, some deviation can be seen for the pitch, and especially the pitch rate, as seen in Figure 5.6, when time grows.

While all the results mentioned are without any noise, Figure 5.7 shows the results when measurement noise is simulated on the system. Notice that the observer is tuned with about half the speed now. That means that the poles are on a semicircle with half the radius of the one seen in Figure 5.4.



(a) Linear and angular positions. Real states in blue and estimated in red.



(b) Linear and rotational velocities in blue, and the estimation in red.

Figure 5.5: LQG simulations.

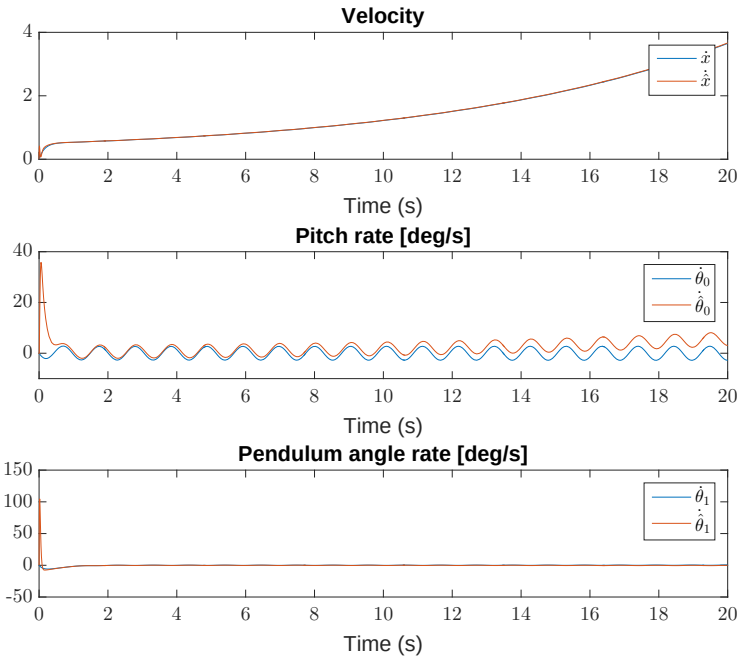
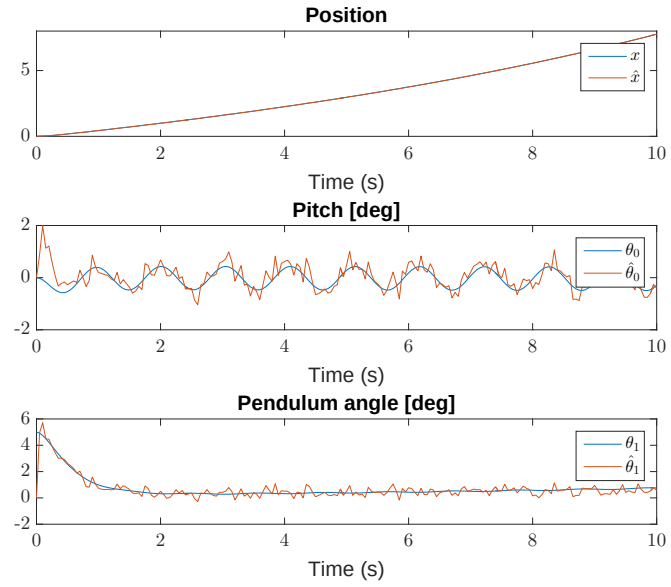
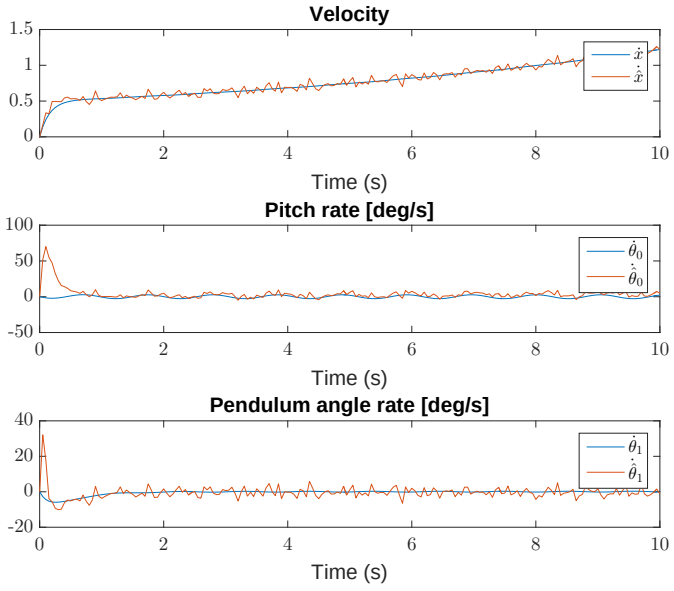


Figure 5.6: LQG simulations over 20 seconds.



(a) Linear and angular position. Real states in blue and estimated in red.



(b) Linear and rotational velocities in blue, and the estimation in red.

Figure 5.7: LQG when measurement noise is added to the simulation.

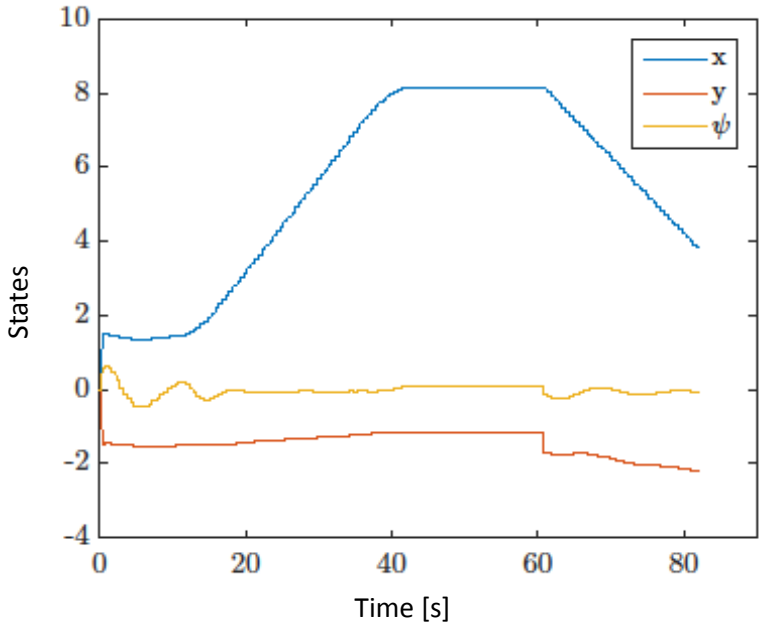


Figure 5.8: Vessel is given a constant commanded surge force $X = 0.5$ N in positive x_b direction. Then reversed with a constant commanded $X = -0.6$ N. The vessel is stationary initially. The yellow line shows the heading.

5.2 Experimental Results

All the results from the model tests carried out in the MC Lab are presented in this chapter, and further discussed in Chapter 6. All tests are carried out with the x_n position artificially set to zero, i.e. position is not controlled.

5.2.1 Heading Controller

The heading controller (4.18) have been implemented and tuned with $K_p = 0.1$ and $K_d = 0.12$.

Figure 5.8 shows a run with constant commanded forward thrust, $X = 0.5$N, then constant negative thrust $X = -0.6$N. Notice that a constant force gives a constants speed as the force rapidly is balanced by hydrodynamic forces. The blue line shows the x_n position of the vessel, while the yellow line shows the heading. The red line is position along y_n.

Doing the same experiment as shown in Figure 5.8 with the heading controller

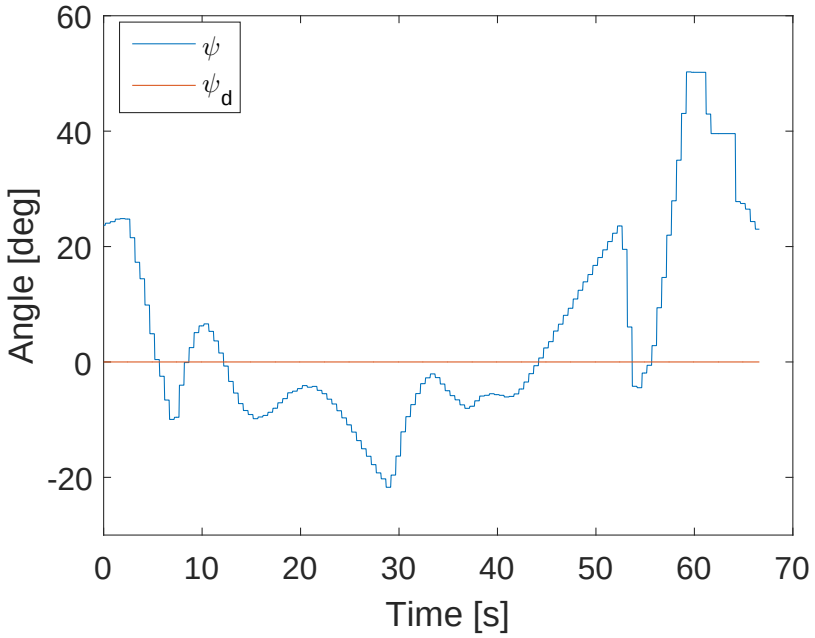


Figure 5.9: Heading when the vessel is commanded to go in a straight line, with the heading controller disabled.

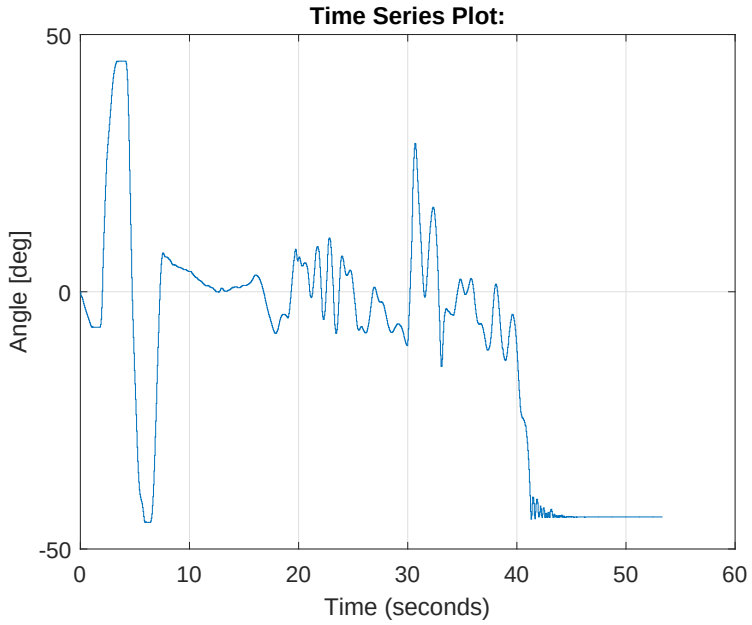
turned of results in the vessel yawing, even tough the only commanded force is strictly along the x-axis. The yaw angle is shown in Figure 5.9.

5.2.2 State Reduced LQR

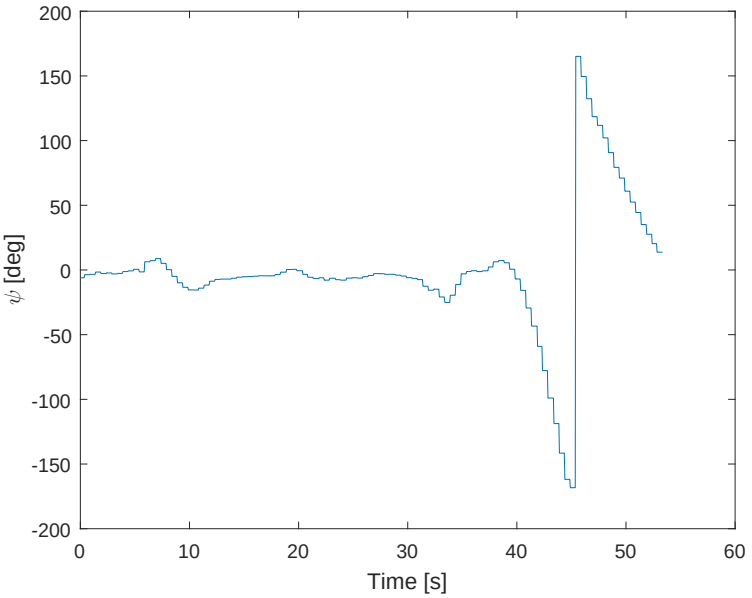
The first laboratory experiment with the marine inverted pendulum was conducted with a simplified LQR controller, based on the cart-pendulum model described in Section 3.2. The state vector is then in $\mathbb{R}^4$, and the A and B matrices are taken as in (3.3), without further coupling effects or hydrodynamic parameters described in Section 3.3 and 3.4. Notice that only the pendulum angle is controlled. Not the position.

Figure 5.10 shows a typical run with the LQR approach described above. The first 10 seconds or so are without controller enabled, as the pendulum is moved to its extremes manually. This is done to find the absolute zero. From about 10 seconds the controller is enabled and the pendulum is initialized close to the vertical upright.

Notice that results presented here are carried out with the springs attached as in



(a) Pendulum angle θ_1 with a simple LQR as if the vessel was a cart. Initially it maintains control, but when pendulum falls over it is not recovered.



(b) Yaw angle is initially kept close to zero, but heading controller fails at large velocities.

Figure 5.10: Simplified LQR Control.

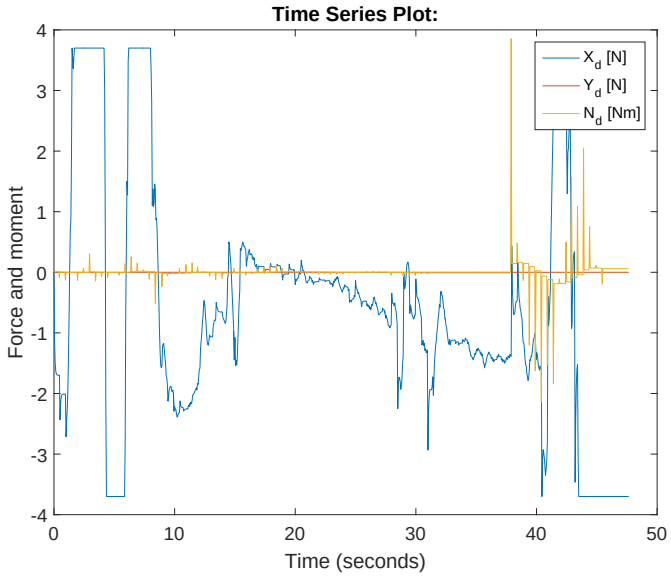


Figure 5.11: Reduced state LQR. Desired force to the thrusters corresponding to Figure 5.10. Notice that the thrusters are kept off during the first 10 seconds or so, to calibrate the pendulum.

Figure 2.2b. That means that they somewhat slows the pendulum dynamics, but the effect is small.

The desired thruster force corresponding to the results in Figure 5.10 are shown in Figure 5.11. The gain matrix for the feedback gain during the runs is

$$K_{\text{LQR}} = [-0.1208 \ \ -16.1172 \ 0.0403 \ \ -6.1839] \ .$$

K_{lqr} is the gain K_c presented in (4.20). Here used on the 4 DOF system presented in Section 3.2.

The Simulink diagram is attached in Appendix A.21.

5.2.3 Full State LQR

In this section a LQR controller have been designed based on the coupled MIP equations (3.31). The springs are kept as described in previous section, and only pendulum angle is controlled. A Simulink diagram is attached in Appendix A.23.

An experimental run, where the feedback gain K_c from (4.20) (now called K_{lqr})

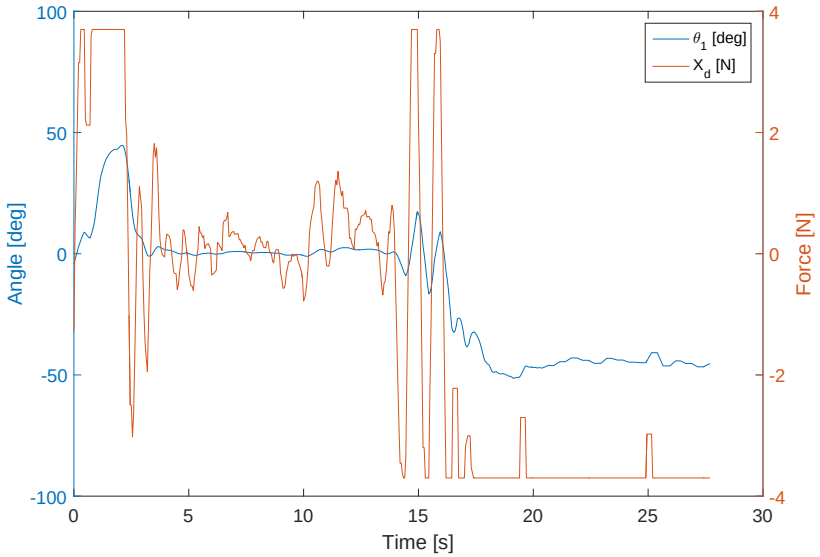


Figure 5.12: Pendulum angle, blue, and commanded thrust force in red.

is

$$K_{\text{LQR}} = [0.0067 \quad -0.5648 \quad -16.0613 \quad 0.0914 \quad -0.5912 \quad -7.2082]$$

is shown in Figure 5.12. The controller is turned on after approximately 5 seconds and the pendulum initialized close to the upright. Initially the pendulum is kept there, with small oscillations in the commanded force in surge. After 15 seconds the pendulum starts to fall, and the commanded force is saturated.

Results from a more aggressive tuning of the LQR, with

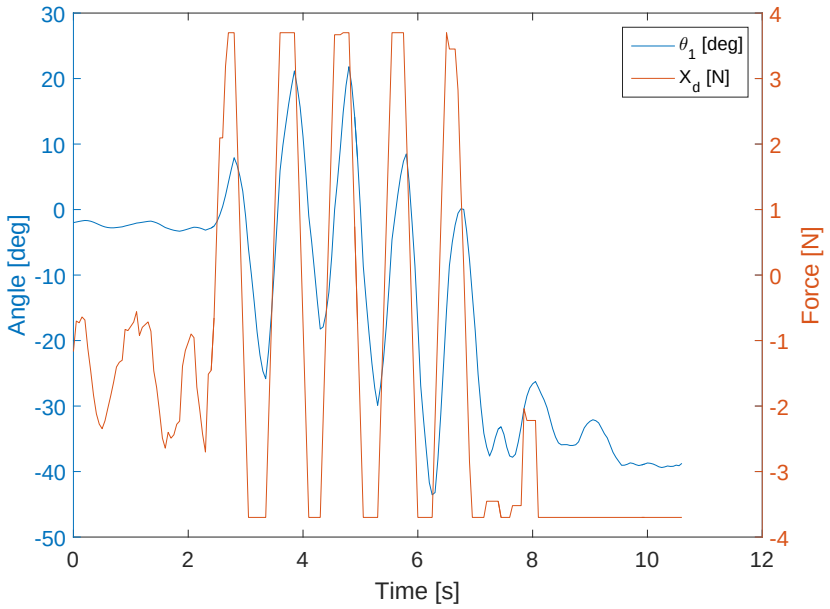
$$K_{\text{LQR}} = [0.0250 \quad -1.1373 \quad -33.7031 \quad 0.3489 \quad -1.5075 \quad -13.4610]$$

are shown in Figure 5.13, and a short video is made available by Sharoni (2016).

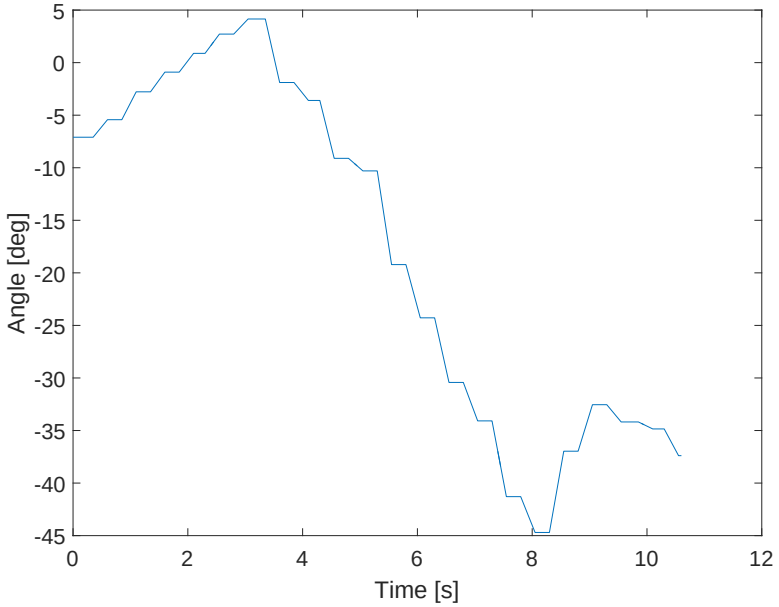
Figure 5.14 shows how the vessel is pitching when controlling the pendulum.

Another typical result can be seen in Figure 5.15. The pendulum is released at close to upright after 5 seconds. It keeps position for a few seconds, then a small oscillation before it falls over. Figure 5.15b shows the heading in this case.

Figure 5.16 shows another experiment with the aggressive tuning of the controller. Figure 5.16b also shows the position along x_n, which is uncontrolled.



(a) Relative aggressive tuning of the LQR shows a pendulum oscillating about the equilibrium, before it falls over. The thrusters change from maximum positive to maximum negative thrust as the pendulum changes side.



(b) Yaw angle corresponding to the figure above.

Figure 5.13: Results with a six state model controlled with LQR.

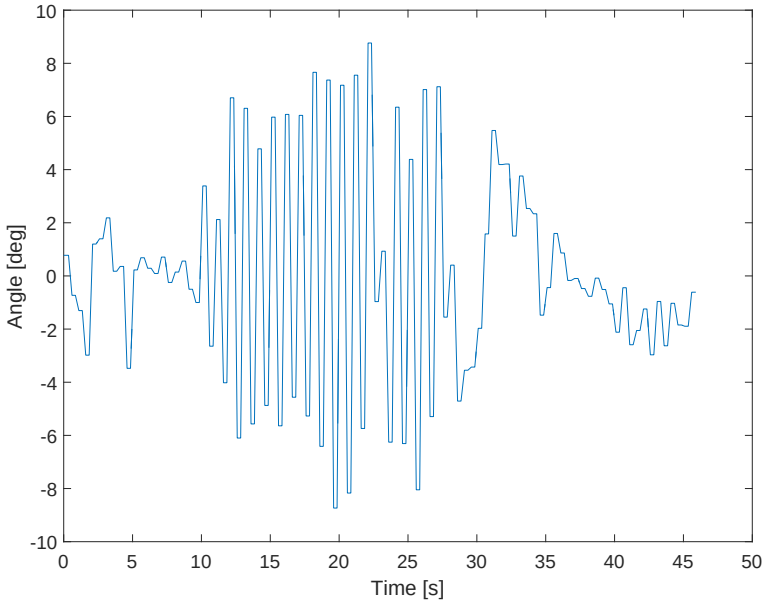
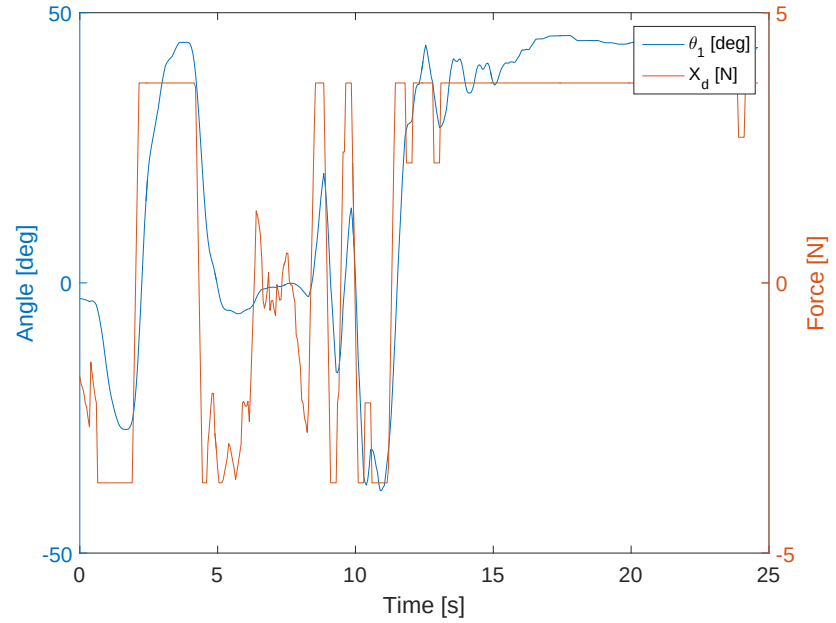
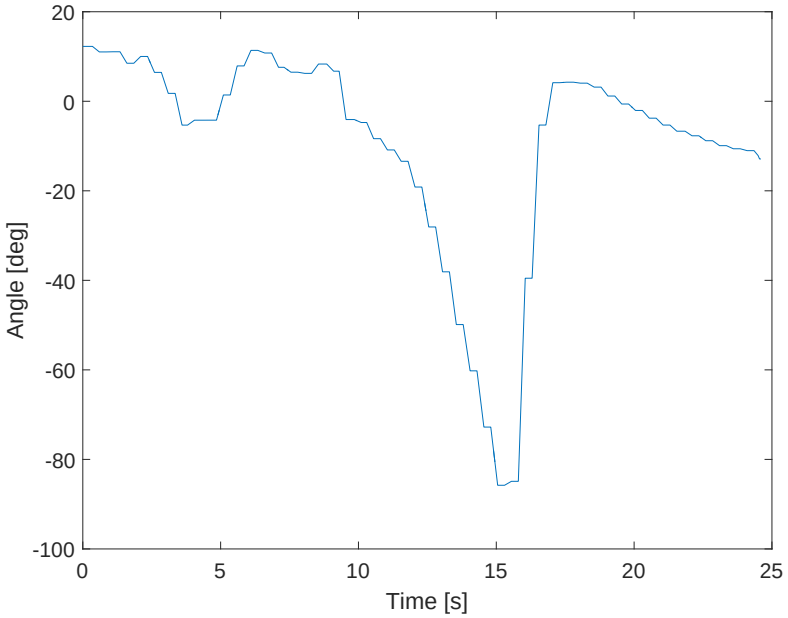


Figure 5.14: Pitch angle corresponding to the results in Figure 5.13.

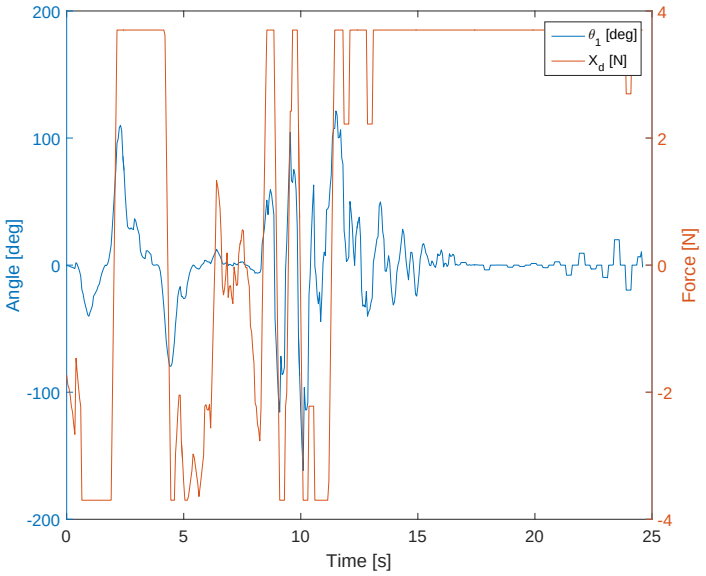


(a) Pendulum is released upright after about 5 seconds, then controlled by the LQR. A typical result from the lab.

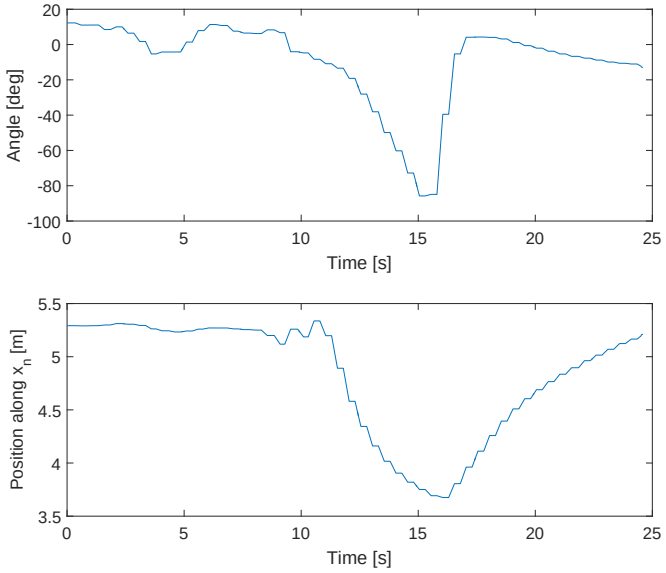


(b) Heading corresponding to the figure above. A typical result from the experiments.

Figure 5.15: Results from a run with the 6 state LQR controller.



(a) Pendulum is released upright after about 5 seconds. Plot shows pendulum angle in blue and thruster commanded force in red. Gains are as given above for the aggressive LQR.



(b) Heading and position along x_n. After 15 seconds the vessel is dragged back manually.

Figure 5.16: Experiment with the LQR, also showing the position, which is not controlled.

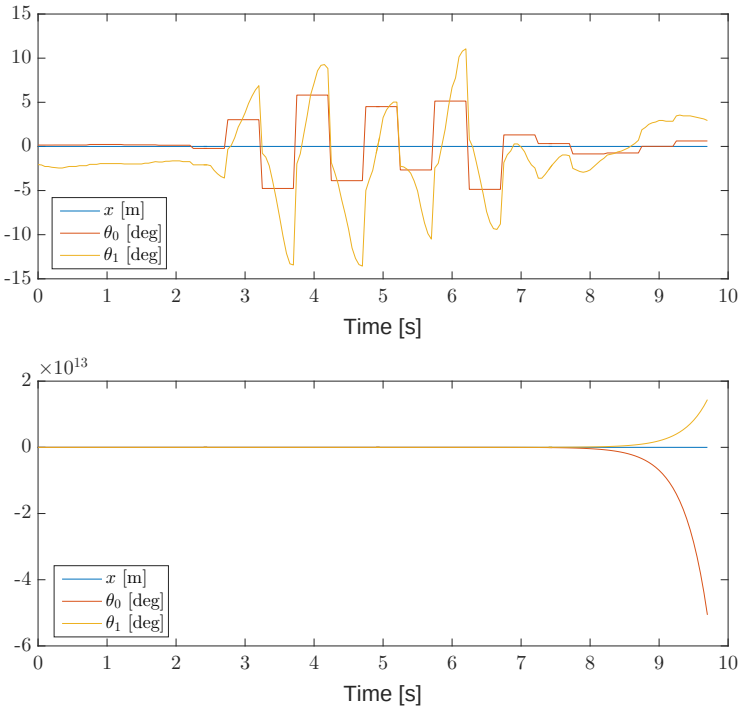


Figure 5.17: Measured position states (top), and estimated position states (bottom). Notice that the position is artificially set to constant zero.

5.2.4 Full State LQG

As an extension to the results presented in Section 5.2.3, several tests have been conducted with a Luenberger state estimator added to the control system. A result is shown in Figure 5.17, where the true position is artificially set to zero. It is clearly seen that the observed states diverge. Observe the scaling of the y-axis. A series of tests have been conducted, some where the position is measured, and not set to zero, and some where the pendulum is artificially kept upright. The estimated states from the Luenberger observer in those cases are more or less equal to those presented in Figure 5.17, and hence not presented here.

Chapter 6

Discussion

6.1 Mathematical Model

Drawing the attention over to Figure 3.18a. The pendulum is falling in the negative direction, which is expected as the vessel bears off in a positive x-direction. However, a closer study of the two lower plots reveals some strange behaviour.

First of all, when the pendulum is falling in a negative direction, it is expected, and experiments show, that the vessel is pitching in the same direction, given the frames of Figure 3.14. However, by looking at Figure 3.18a it is seen that the vessel gets a positive pitch angle. Furthermore, the thruster force results in a negative pitching moment. Together this should add up to give a noticeable negative pitch angle. It is clearly seen that this is not the case, and hence should be further investigated.

The second issue that will be addressed is found by observing the lower plot in Figure 3.18a and 3.19. It is clearly seen that the pendulum (in blue line) behaves as expected. However, the red line showing the angular speed does not seem to behave accordingly. From both figures, it seems that the angular speed is zero, but since speed is the derivative of position, it can not be zero when position is changing. If one zooms in, the figures shows that the angular speed indeed is different from zero. Nevertheless, it is much smaller than it should be. For instance, in Figure 3.19, the pendulum rotates through 360° in about 3-4 seconds. This should give an angular speed close to 100 deg/s, which clearly is not the case in the simulations.

It is important to mention that although the mathematical equations are considering all the coupling and hydrodynamic effects, the friction in the hinge, as well

as the springs are left unmodelled. They do indeed change the behaviour of the system, and might be a reason for mismatches between the true and theoretical model.

6.2 Controller and Observer Designs

The mentioned concerns regarding the correctness of the mathematical model directly affects the simulation results where controllers and observers are tested. Throughout the project the equations presented in Section 3.4 are used to design controllers and observers. However, as will be commented, questions regarding the controllers and observers are also related to the mathematical model of the system.

The LQG approach have been chosen for the experiments since it is well proven to be a good approach stabilizing inverted pendulums. Moreover it is well known to the author, and the cost function approach is easy to tune as it gives a very rational understanding of the controller.

The following sections presents interpretations of the results, both from computer simulations and experiments in the MC Lab.

6.2.1 LQR

For this specific system, with state vector z being the three positions and three velocities, as described for (4.1), the LQR is the same as a PD controller. That is, a PD controller is some constant gain multiplied with the position state, and another constant gain multiplied with the derivative of the first state. This is then exactly what is happening in the LQR. Hence, there is no need to design a PD controller for the system, as the LQR approach is a smart way of doing the same.

It is expected that the uncontrolled system is highly unstable. This is clear from Figure 5.1a showing that five poles are in the right half complex plane, and one is on the imaginary axis, and hence marginally stable. However, for the closed loop system, Figure 5.1b shows all the poles being at least marginally stable. It can be remarked that the system is a minimal realization, and hence the poles are equal to the eigenvalues.

It is clearly seen from Figure 5.2 that the closed loop system have an unstable behaviour, as the states seems to initially stay close to the desired reference, but then bears of and oscillates after about 25 seconds.

So, based on the above statements about the poles, it is hard to explain why the system becomes unstable, especially since it seems to behave in a controlled way initially.

It is even harder to give a good interpretation from Figure 5.3. The system seems to behave as if no control action is applied at all. The pendulum swings back and forth within 360°, causing the vessel to pitch and oscillate around the initial equilibrium state. Changing the weight matrices Q and R does not influence the results at all. It does influence the control input, but not the states of the system.

6.2.2 Luenberger Observer

Tuning the observer can be a challenging task. The desired location of the poles is not straight forwardly related to the physics of the system. The poles are then chosen to achieve fast dynamics, as the observer must be faster than the controller, at least. However, if chosen too fast it can be unstable. Sharoni (2015, Sec. 4.4.2) included a discussion on how to place the observer poles, mainly based on Chen (2013).

In this case, the poles seen in Figure 5.4 are located with a relatively large radii, approximately 20 times the most negative closed loop controller pole. This is to make sure the observer is fast enough to capture the dynamics of the system. This, in combination with the spacing between the poles results in some overshoot, as seen initially in Figure 5.5. However, this tuning of the observer results in a very good state estimation.

With time, however, the estimated pitch, and especially pitch rate, deviates from the true states. This is illustrated by Figure 5.6. This is probably mainly due to the fact that the true model is highly nonlinear, where the observer is based on a linearized system. However, as mentioned, there is some uncertainty connected to the correctness of the equations, so this can also influence the results.

When measurement noise is added, as shown in Figure 5.7 the observer must be tuned less aggressively. That means that the gain L must be reduced in order to rely more on the model, and less on the measurements. In this case the poles are placed on a circle with half the radius of the tuning without noise. As shown, there is still some noise on the estimates, especially for pitch. This effect can be reduced by decreasing L further, but can result in the observer drifting from the real states, or not being fast enough.

The discrete time Kalman filter covered in Section 4.2.2 have also been simulated, as attached in Appendix A.10 and A.11. However, the results do not differ from the ones with the Luenberger observer, and thus the simulations are not presented here.

6.3 Experimental Results

Since both the encoder, and Qualisys system measures position only, the three derivative states, i.e. the velocities are unknown.

Now, mathematically the velocity for each state can simply be found by taking the derivative of the position states. However, in reality the measurements can be noisy and unreliable. Moreover the measured positions might be digital, and hence not continuous. This will result in problems if they are differentiated.

In spite of this, the Qualisys system returns very precise data, without noise. The same goes for the encoder. Therefore, "dirty derivatives" have been used in the laboratory experiments, simply differentiating the position measurements by using derivative blocks in Simulink.

6.3.1 Heading Controller

As mentioned in Section 5.2.1, the user controls the vessel by setting desired thrust force. Since the thruster forces are balanced by hydrodynamic forces, a constant force results in a constant speed, after a short period of acceleration.

It is seen from Figure 5.8 that there are some oscillations in yaw angle initially, while the vessel is stationary. However, when the vessel moves the heading controller works satisfactory, keeping the heading at zero degrees. There is some misalignment in the y position, meaning it does not stay at zero.. This is since the body frame is not perfectly aligned with the basin frame. However, the heading controller is controlling yaw in body frame, and does indeed keep the yaw angle at zero.

From Figure 5.9 it is very clear that the vessel is rotating quite uncontrollably when the heading controller is disabled. Notice that this figure is under the same conditions, and thruster inputs as Figure 5.8. The only difference being that the controller is disabled. This then clearly illustrates both the need for, and functionality of the heading controller.

However, as seen from Figure 5.10b, 5.13b, 5.15b and 5.16b, the heading controller does not seem to work well at large commanded thrust. This might be since all available thrust is used trying to bring the pendulum to the upright equilibrium, not leaving enough in spare for the heading controller. Another reason is that the heading controller is tuned relatively slow, as it easily becomes unstable. Hence, the controller simply is not fast enough to cope with the rapid yaw rates at large thrust.

6.3.2 State Reduced LQR

The state reduced LQR has been applied to the system since it is interesting to examine whether a more complicated model really is necessary.

First of all, Figure 5.10a is a good example of how the system must be initialized the first time power is connected. The pendulum is manually brought to the two extremes in order for the system to know where the centre, i.e. upright is.

The same figure also shows a typical result. The pendulum is initialized very close to $\theta_1 = 0°$. It is kept there, slightly oscillating, but when it falls too much to one side, it is not recovered. The corresponding thruster input in Figure 5.11 shows how the desired force in X slowly increases as the pendulum starts to fall over to one side, and then is saturated after it has fallen.

From this it seems like the desired thrust is not increasing rapidly enough to cope with the falling pendulum. The reason for the pendulum leaning towards one side, but not falling initially is due to the support from the springs. It is a clear tendency in commanded force, trying to compensate for the falling pendulum. However, since the force is slowly increasing, it never supplies the sufficient acceleration needed to tilt the pendulum back.

Moreover, the linear model is only valid within a certain region, close to $z = 0$. The more some of the states starts to deviate from this region, the more inaccurate the model becomes. Consequently the model based controller is strongly affected resulting in a weakened performance.

Figure 5.10b and the yellow line in Figure 5.11 shows how the heading is kept very close to zero, as desired, when the pendulum is upright. However, the vessel rotates strongly once the pendulum falls over too much, and the thruster force increases. This indicates that the heading controller does not work for large commanded thrust.

Notice also that the although the desired state is all zero, only the pendulum angle is controlled. This is achieved by artificially setting the other two states to zero, instead of using the measurements for feedback.

6.3.3 Full State LQR

Now the LQR has been designed based on the coupled equations (3.31), and is hence applied to the full 6 DOF model. The hydrodynamic parameters are not included due to large uncertainties in the model at the time of the experiments. As in previous section, the position is artificially set to zero, i.e. it is not controlled.

Figure 5.12 shows a very typical experimental result. The pendulum is released close to the upright equilibrium. The controller then is able to maintain the pendulum balanced, with small oscillations in commanded thrust. However, after about 15 seconds the pendulum falls too far over to one side. The result is a large commanded thrust, bringing the pendulum back to the other side. This again results in a equally large thrust in opposite direction to compensate, resulting in the pendulum falling back over again. Finally this leads to the pendulum falling over, not being recovered.

While the scenario described might indicate that the controller is to aggressively tuned, experiments does not support that conclusion. Reduction in the feedback gain only leads to the pendulum falling over right away, as the controller is not compensating enough once it starts to loose balance. These results are not included, as they are all very short runs showing the pendulum falling over immediately. Nevertheless, an example is given in Figure 5.15.

Consequently, another approach is tried, where the controller is tuned with an even larger feedback gain. The aim is then that it will give a large thrust force once the pendulum starts to fall, which is enough to force the pendulum back in the other direction. Results from this approach are given in Figure 5.13, and in the short video by Sharoni (2016).

This tuning of the LQR does indeed give the results expected. As seen from Figure 5.13a. The pendulum falls back and forth, but is kept oscillating about the equilibrium. The corresponding thrust goes from maximum to minimum, saturating the thrusters. This is an unstable controller, and the pendulum does indeed fall over eventually. Furthermore, this approach causes a huge amount of stress, and wearing on the thrusters, as will be further elaborated in Section 6.4.

As mentioned, the main control objective is to balance the pendulum, and therefore position control is not implemented. The lower plot in Figure 5.16b shows the true position of the vessel along x_n. As seen, position is more or less kept while the pendulum is upright, and then once it falls over, the vessel bears of at maximum speed in the same direction, trying to recover the pendulum, which it does not

manage to do. After 15 seconds the vessel is dragged back manually resulting in the position to going back to start.

6.3.4 Full State LQG

The Luenberger state estimator described in Section 4.2.1 have been implemented (see Appendix A.25) and tested in the MC Lab. After several runs, the results did not improve from the ones shown in the lower plot in Figure 5.17. As seen, the estimated states converges and hence the Luenberger observer designed in Section 4.2.1 does not work.

There are several possible reasons for this. Obviously, the Luenberger observer as a design approach is well tested, and sucessfully applied to several applications. It was also shown in Section 5.1.2 that it indeed works for simulated examples. So, the reasons for it not working in the laboratory might be due to bad tuning. The desired location of the poles might not be feasible, and the observer gain may be too large, causing the observer to be unstable. However, there is reason to believe that there are other causes to the divergence of the observer states.

First of all, the Luenberger observer is designed based on the linear system model (4.2). The real system, however, is highly nonlinear and hence this can cause the observer to be too far off. Moreover, the linearized model is highly sensitive to system parameters and modelling errors, which one have to expect are present.

When the position for the pendulum or vessel are set to constants, the observer also fails. This is not too unlikely, as the observer model "expects" the system to behave in a certain way, but when the measurement arrives it does not correspond at all with the expectations.

Hence, it is believed that it is a question of design method, and not tuning. Therefore, some suggestions follows in Section 7.2.3 on how to implement an observer that sucessfully gives usable state estimates in the laboratory.

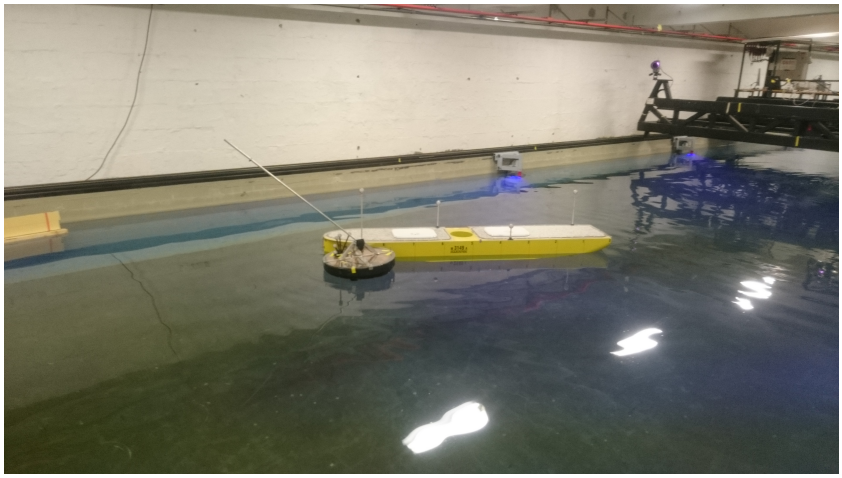


Figure 6.1: Things do not always go according to plans. The CS Inocean Cat I Drillship pushing the CS Saucer back ashore.

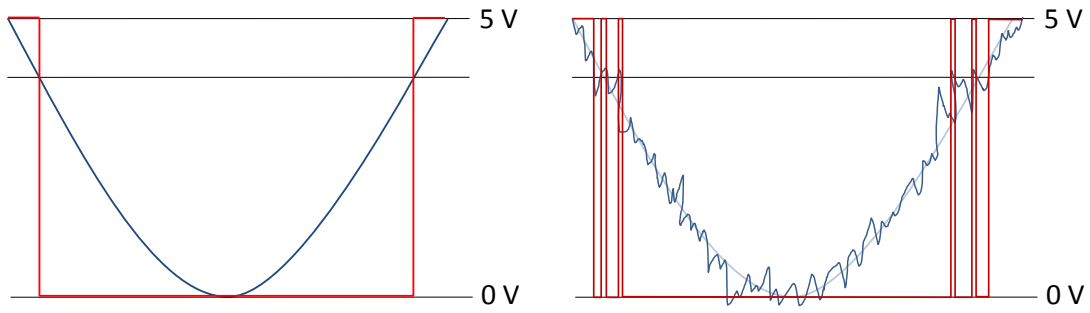
6.4 General Remarks

As illustrated in Figure 6.1, things does not always go as intended. There are many reasons for this. Some will be further reflected on in the following.

First of all, as clearly illustrated by Figure 5.14, the vessel is pitching significantly with increasing thrust force. This is due to the configuration of the vessel and thrusters, as explained in Section 3.3. This significantly complicates the equations for the system, and the control objective. If, on the other hand, no pitching occurs when the vessel is subjected to thrust, the system starts to resemble the cart-pendulum system. This is, as explained in Section 3.2, a much easier control system. The hydrodynamics would still play a role, but the equations would be simpler.

So, although it would be a somewhat different project, the best equipment for a marine inverted pendulum would probably be a very long and slim vessel. One would expect a greatly reduced pitching effect, and furthermore the hydrodynamic damping in surge would decrease, giving faster response. This would also eliminate the need for a heading controller, as such a vessel would be very directionally stable.

Another issue is the thrusters, and thruster control. Figure 5.13a clearly shows how the thrusters are commanded to go from maximum negative to maximum positive thrust in the fraction of a second. Although the thruster response is really quick, a reference model should probably be made before the desired command is



(a) Ideal analog signal in blue, and digital interpretation in red.

(b) Real signal contains noise, which causes the digital signal to oscillate rapidly around the threshold.

Figure 6.2: Signal as the magnet passes the hall effect sensor, for rpm measurements.

sent to the thrusters. The reference can be a second order transfer function, reflecting the physical limitations of the thrusters. This model could be tuned by measuring the actual thruster response. The rpm measurements from the hall effect sensors are very useful in that sense, but they were not installed at the time of the experiments.

So, the thrusters do seem to cope relatively well with the very rapid changes in force and direction, however, this way of running the thrusters causes a lot of wear and tear. Moreover, some instability and unreliable behaviour have been noted from the thrusters during experiments. This effect could probably be reduced by sending a more feasible command to the thrusters. Nevertheless, the control objective do require a very agile vessel, with rapid changes in thruster force and direction. Hence, (reducing) wear and tear on the thrusters is not a focus in this project.

The rpm measurements are to noisy to use for feedback. This is illustrated by Figure 6.2. Figure 6.2a shows an ideal analog signal in blue, and the digital interpretation in red. The digital signal is set to 0 V when the measured analog signal passes some threshold. However, the real measurements are noisy, as shown with the blue line in Figure 6.2b. This causes the signal to pass the threshold back and forth just around the intersection points, causing the digital signal to shatter as shown by the red line. This is called hysteresis. So, the measurements are good when shown on a digital display, however to noisy to use for feedback.

It is also worth mentioning that all the experiments were conducted with the supporting springs attached at an angle, as shown in Figure 2.2b. This give some, but not too much, support. Experiments without springs resulted in the pendulum falling over too fast for the vessel to recover it. That mentioned, the springs also caused some problems. For instance, it resulted very tricky to adjust both springs equally, resulting in a tendency for the pendulum to fall more easily over to one side. This affects the results, and moreover complicates the task for the controller, as more force is required to recover the pendulum from one side compared to the other. Furthermore, the effect of the springs is not modelled at all. As the LQR is a model based controller, it is relatively sensitive to model parameters and modelling errors.

As mentioned above the vessel is pitching significantly under experiments. This results in water splashing over the deck, but thanks to the plexiglass lid, and watertight seal shown in Figure 2.18, the electronics are kept away from water and not damaged. Hence, this proved to be an important modification.

A great deal of time was spent installing and tuning hardware, and on troubleshooting. It turned out that there was disturbances and noise, affecting the system and causing unreliable behaviour and glitches in the thrust. This was fixed by attaching ferro magnets to all the wires providing PWM signals to the thrusters.

It was discovered that the friction in the pendulum joint is noticeable, and might be too large to neglect in the equations. This might cause the controller to fail, as a larger amount of force is needed to overcome the friction, and change the position of the pendulum.

Although not included in Chapter 5, experiments with a longer pendulum were also carried out. Sharoni (2015) showed how a longer pendulum would have a lower frequency, and hence should be easier to control. The pendulum should be made as explained in the final part of Section 2.1. However, since such a thinner or thicker rod were not available, a solid aluminium rod were forced inside the pendulum, thus making it about 70 cm longer, but also significantly heavier. Experiments with this setup did not show any improvements, in fact, the system behaved worse, almost causing the vessel to tip over when the pendulum fell.

So, it is believed that a longer pendulum will be easier to control, but it will have to remain light. Both this experiment, and results with added point loads on the top of the pendulum shows a poorer behaviour compared to the original pendulum.

Chapter 7

Conclusion and Further Work

7.1 Conclusion

The vessel has been heavily upgraded during this project. Wiring has been re-done, adding heat shrink and better wire terminations. Moreover, several components from the original design presented by Idland (2015) have been removed, as they proved unnecessary. One of the main issues has been the unstable thrusters, due to interference. The problem was solved by applying ferro magnets around the wires leading PWM signals to the motors.

An entirely new software platform have been developed, based on ROS. The system implemented by Idland (2015) was based on National Instruments myRIO (National Instruments, 2015) and is both more expensive, and less flexible, in the authors opinion. The new system based on ROS uses an Arduino embedded circuit board (Arduino, 2016) and a Raspberry Pi 2 (Raspberry Pi Foundation, 2016). They are both low cost items, and have a huge community online which makes them easy to use. Moreover, the main advantage with ROS for this project is the ease in which it interfaces with MATLAB and Simulink.

The ROS platform has proven easy to work with during the laboratory experiments. It is indeed very flexible, as one can work directly on the RPi2 as well as on the computer. Files can simply be copied and pasted between the units. New sensors are easily implemented in the Arduino code, as the majority of sensors already have a written code found online. Furthermore, altough not fully used in this project, ROS has its own graphical user interface, and can generate a lot of graphs, tables and figures with one simple command.

Hence, it is believed that this change of platform and embedded controller really

serves future projects. The CS Saucer is now both more easy to use, and provides great flexibility regarding upgrades and sensor changes for future projects.

A pendulum payload system have been designed and constructed. It is a strong construction, but still relative light in weight, adding up to 1,05 kg for the whole construction, including the encoder. As a comparison, the plexiglass lid alone is 1,14 kg.

The drawbacks with the construction is that it only allows rotation in one plane, thus not allowing to easily expand and complicate the problem by allowing the pendulum to rotate in several planes. Moreover, although it is light, it is still a bit on the heavy side.

In Chapter 3 one of the main contributions from this thesis is presented, namely the derivations of the system equations. Since equations for both inverted pendulums, and marine vessels are well developed in the literature, the problem is attacked by examining each part individually. The challenge lies in combining them. In this sense, the double inverted pendulum have been a good source for comparison.

The equations are derived using the Lagrangian approach. However, Newton's laws could also be applied, but results in a more messy derivation. Furthermore, it is shown how the system can be considered as an industrial manipulator. The advantage of doing this is that there exist a huge variety of control algorithms for such manipulators, and moreover that the Lagrangian is already solved in the literature.

For each step along the road, simulations are carried out to verify the correctness of the equations. However, as pointed out in Section 6.1, there is reason to believe that the derivations should be revised.

The equations are highly coupled and nonlinear. However, for control purposes they can be linearized around the desired equilibrium point. The linear equations can then be used to write the system on state space form. From there, a LQG control system is designed. This is a linear quadratic regulator with a Luenberger state estimator. The LQG is chosen as it is widely used, and proven to be a good choice for inverted pendulums. That said, there exists other control algorithms that might be suitable, which are proposed, but not implemented.

Since an accuare model is important to design good controllers and observers, the main focus has been on developing a good mathematical model. Nevertheless, it should be mentioned that the model parameters used in the model are taken as found by Idland (2015). They are not accurate for the system with the pendulum payload, and moreover not all paramers are known, and hence qualified guesses are used. This directly affects the control and observer, as they are model based,

and hence highly sensitive to model parameters.

The thesis headed off by asking whether it is possible to maintain an inverted pendulum upright on a marine platform such as the CS Saucer. It is not successfully shown in this thesis. Nevertheless, the results are showing promising tendencies. So, to answer the main question and conclude, from the findings of this thesis it is believed that it is possible to achieve control of the inverted pendulum on a marine platform. It is also believed that this can be done using the CS Saucer, but it will require some more effort and modifications. A reflection on those follows in Section 7.2.

7.2 Further Work

Implementation of the new software platform, as well as further modifications and sensors added to the CS Saucer, results in a versatile platform that can be used to continue this project, as well as other projects. Hence, recommendations for further work follows in this section.

7.2.1 CS Saucer

There are two main issues regarding the CS Saucer used as a platform for the inverted pendulum; the vessel is very directionally unstable, and the thrusters are somewhat unreliable. The heading can, as shown, be stabilized by control. However, while the PD controller used in this project performs well at low to moderate thrust forces, it fails at high forces. Therefore, more effort should be put into heading control. Notice that this does not need to be a question of tuning or control design alone, it can also be done by thrust allocation and configuration, and by making physical modifications to the vessel. For instance, since only back and forth movement is needed for this project, fitting a fin to the vessel is a good and cheap solution.

The other main issue mentioned is that the thrusters are unreliable. With this it is to be understood that thrust does not increase or decrease in a smooth way. Furthermore, glitches are experienced where sudden jumps in thrust can be detected. There is also a problem when the thrusters very rapidly changes direction of rotation.

One step is to improve the mapping from desired force to the PWM signal given to the motor. In collaboration with Ueland (2016) this have been done, and the results are presented in their thesis. The new mapping showed improvements, but further work needs to be invested here. For instance, rpm control can be implemented with feedback, as rpm measurements are available. However, some filtering must first be considered as the measurements are noise. A RC-circuit is believed to be an appropriate way to do this. Since the rpm is read via the Arduino interrupts, it is hard to design a low pass filter in the software that works satisfactory.

Finally, the mapping from thrust to PWM is assumed equal in both directions. In reality, however, there is a difference depending on direction of rotation. This results in a bias from the two rear thrusters, since they are counter rotating.

The thrusters also show unstable behaviour when rapidly changing thrust direc-

tion. This is also related to the mapping, as there is a "dead zone" around zero thrust. Nevertheless, improvements could possibly be made by changing components, as motors and speed controllers.

A final note regarding the thrusters is that a more advanced thrust allocation should be considered for future projects. By that it means that one should take advantage of the rotational property of the thrusters to achieve better directional control. However, that would result in a highly over actuated vessel, and hence a non linear optimization problem must be solved to find the thrust allocation.

Regarding the hardware layout in the CS Saucer, one should make some racking system for the battery, Arduino and Raspberry Pi to make sure the wiring is kept tidy. Finally, a setup with four equally spaced thrusters should be considered, as it is believed to improve the symmetry and hence directional stability.

7.2.2 Pendulum Control

This thesis concluded that although no complete success was achieved balancing the inverted pendulum, it seems feasible, given some modifications. Some are already mentioned in Section 7.2.1, but more specific recommendations follows here on the pendulum control.

First of all, experiments without supporting springs revealed that the vessel, or control system, is to slow to react to the fast pendulum dynamics. It is believed that two main steps can be taken to overcome this. First of all, it is proposed to experiment with a longer pendulum. Then, it is highly recommended to take steps as described in previous section, as the main delay lies in the zone close to zero thrust, when the thrusters change direction of rotation. Notice that if a longer pendulum is used, the weight should be kept low, as it was shown in this thesis that a longer, but heavier pendulum did not give good results. The author proposes using a carbon rod, e.g. a cross country ski pole as the pendulum.

An interesting approach that should be tested is to constrain the minimum thrust in both directions, to say 30 % of max. The vessel should then remain stationary, as the two forces in each direction should cancel each other. Then, one can reduce or increase thrust in either direction to control the pendulum, thus avoiding the problematic zero thrust region, and shifting direction of rotation. Rudaa et al. (2016) shows success using this approach for roll damping of ships using conventional thrusters. Notice that if this approach is chosen, a setup with four thrusters would be much more applicable.

Rudaa et al. (2016) also propose a nonlinear modification to the PD controller,

where the derivative gain is multiplied with a factor

$$\text{sign}(\dot{\theta}_1) \cdot \sqrt{\frac{|\dot{\theta}_1|}{c}} \ ,$$

where c is a tuning factor. In this way, the term increases when $|\dot{\theta}_1| < c$, thus resulting in higher control output. Conversely the term decreases for $|\dot{\theta}_1| > c$, avoiding saturating the thrusters for large pendulum velocities. This strategy yields good results for the roll damping application, and should also be applicable for the pendulum controller.

Now, mentioned in the thesis is the correctness of the mathematical model. This model should be further revised, and model parameters should be refined to increase the accuracy. This means, for instance, that the contribution from the springs, and friction in the hinge should be included in the mathematical model. If the mathematical model is improved, given it is inaccurate, and the model parameters match the real vessel, a huge improvement is expected from the controllers and observers, since they are model based.

7.2.3 Future Projects

Beside the recommendations mentioned in Section 7.2.1 and 7.2.2, there are some other modifications, and proposal for future projects that might be of interest.

First of all, it is proposed to implement other control laws than those used in this thesis. Some alternatives are proposed in Section 4.3. It is believed that a backstepping controller, and the SDRE are especially suitable.

The state estimator certainly needs further work. A test were conducted where the nonlinear passive observer presented by Fossen (2011, Ch. 11.4) were used successfully to estimate the state of the vessel. An idea is to combine this with an observer for the pendulum dynamics alone, thus creating two observers that are very good at each task, and combining them to make a cascade state estimator.

To improve the control law, one could also mount a small accelerometer on top of the pendulum. That way, one could include acceleration feedback, or feedforward in the controller to achieve better and faster control. This was briefly discussed in the project thesis by Sharoni (2015).

Given that the goal for this project is achieved, interesting extensions include controlling the position of the vessel as well as the pendulum. For instance, on could implement path following while controlling the pendulum. The controller should then be extended to at least ten DOF, where the y-position and yaw angle

of the vessel, as well as their derivatives, are added to the six states used in this thesis.

Also interesting would be to test the pendulum control in waves. The mathematical model of the system is still valid, but one could exploit that one can in fact control pitch to some extent, given the large moment arm to the thrusters.

Finally, the pendulum constraints can be relaxed, allowing it to move freely in all directions. This is already achieved for quadcopters, as shown by for instance Hehn and D'Andrea (2011).

Bibliography

Amdahl, J., Endal, A., Fuglerud, G., Hultgreen, L. R., Minsaas, K., Rasmussen, M., Sillerud, B., Sortland, B., and Valland, H. (2011). *TMR4100 - Marin Teknikk Intro. TMR4105 - Marin Teknikk 1. [TMR4100 Marine Technology Intro. TMR4105 Marine Technology 1]*, volume 1. Marin Teknisk Senter, NTNU, 4th edition.

Arduino (2016). *Arduino UNO.* Retrieved 31st of March, from https://www.arduino.cc/en/Main/ArduinoBoardUno.

Bjørnø, J. (2016). *Thruster-assisted Position Mooring of C/S Inocean Cat I Drillship.* Unpuplished master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Blakelock, J. H. (1965). *Automatic Control of Aircraft and Missiles.* John Wiley & Sons, Inc. New York.

Bogdanov, A. (2004). *Optimal Control of a Double Inverted Pendulum on a Cart.* Technical report, Department of Computer Science & Engineering, OHSU.

Chen, C.-T. (2013). *Linear System Theory and Design.* Oxford University Press, international 4th edition. New York.

D'Andrea, R. (2013). *The astounding athletic power of quadcopters.* Seen 10th of October 2015 on https://www.youtube.com/watch?v=w2itwFJCgFQ.

DNV GL AS (2016). *Wadam Software for Marine Hydrodynamics.* Retrieved on 12th of April 2016, from https://www.dnvgl.com/services/frequency-domain-hydrodynamic-analysis-of-stationary-vessels-wadam-2412.

Egeland, O. and Gravdahl, J. T. (2002). *Modeling and simulation for automatic control.* Marine Cybernetics, Trondheim.

Faltinsen, O. M. (1990). *Sea Loads on Ships and Offshore Structures.* Cambridge ocean technology series. Cambridge University Press, Cambridge.

Fossen, T. I. (1994). *Guidance and Control of Ocean Vehicles*. Wiley, Chichester.

Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd.

Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (2010). *Feedback Control of Dynamic Systems*. Pearson, Upper Saddle River, N.J, 6th edition.

Hassanzadeh, I., Nejadfard, A., and Zadi, M. (2011). *A Multivariable Adaptive Control Approach for Stabilization of a Cart-Type Double Inverted Pendulum. Mathematical Problems in Engineering*, 2011.

Hehn, M. and D'Andrea, R. (2011). *A flying inverted pendulum*. pages 763–770. IEEE International Conference on Robotics and Automation, May 2011.

Idland, T. K. (2015). *Design, Construction, and Control of Marine Cybership "C/S Saucer"*. Master's thesis, Norwegian University of Science and Technology. Trondheim, Norway.

Krishnavedala (n.d.). *Cart-pendulum*. via Wikimedia Commons. Retrieved 21st September 2015, from https://upload.wikimedia.org/wikipedia/commons/0/00/Cart-pendulum.svg.

Lundberg, K. and Barton, T. (2009). *History of Inverted-Pendulum Systems*. volume 8, pages 131–135.

Marintek (2016). *ShipX*. Retrieved on 12th of April 2016, from https://www.sintef.no/en/marintek/software/maritime/shipx/.

National Instruments (2015). *User Guide and Specifications. NI myRIO-1900*. Retrieved 27th of October 2015, from http://www.ni.com/pdf/manuals/376047a.pdf.

National Instruments (2016). *Test, Measurement, and Embedded Systems*. Retrived 30th of March 2016 from http://www.ni.com/en-no.html.

NTNU (2016). *Marine cybernetics laboratory handbook*.

Ntnu.edu (2015). *MC-lab - IMT - NTNU*. Retrieved 13th of October 2015, from https://www.ntnu.edu/imt/lab/cybernetics.

Ogata, K. (2010). *Modern Control Engineering*. Pearson, Boston, 5th edition.

Pragues Segway (2015). *Prague Segway Tours*. Retrieved 9th of November 2015 , from http://praguesegway.com/faqs.php.

Raspberry Pi Foundation (2016). *Raspberry Pi 2 Model B*. Retrieved 31st of March, from https://www.raspberrypi.org/products/raspberry-pi-2-model-b/.

REX Controls (2016). *Furuta Pendulum*. Retrieved 10th March 2016, from https://www.rexcontrols.com/furuta-pendulum.

Roberge, J. K. (1960). *The Mechanical Seal*. Bachelor's thesis, Massachusetts Institute of Technology.

Rottmann, K. (2010). *Matematisk Formelsamling. [Mathematical Formulas]*. Spektrum forlag, 12th edition.

Rudaa, S., Steen, S., and Hassani, V. (2016). *Use of Conventional Thrusters For Roll Damping of Ships*. Article in press, 10th IFAC Conference on Control Applications in Marine Systems (CAMS 2016).

Sharoni, R. (2015). *Marine Inverted Pendulum - Preparing for Control using the CS Saucer Platform*. Unpuplished project thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Sharoni, R. (2016). *Marine Inverted Pendulum*. Seen 3rd of June 2016 on https://www.youtube.com/watch?v=y7Glv8lAnwo&feature=youtu.be.

SIMO, p. t. (2013). *SIMO - Theory Manual Version 4.0 rev. 3*. Marintek.

Skjetne, R., Smogeli, Ø. N., and Fossen, T. I. (2004). *A Nonlinear Ship Manoeuvering Model: Identification and adaptive control with experiments for a model ship. Modeling, Identification and Control*, 25(1):3–27.

Spong, M. W., Hutschinson, S., and Vidyasagar, M. (2006). *Robot modeling and control*. Wiley, Hoboken, N.J.

Sørensen, A. J. (2013). *Marine Control Systems. Propulsion and Motion Control of Ships and Ocean Structures*. Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway, 3 edition.

subCULTron (2015). *Submarine Cultures Perform Long-Term Robotic Exploration of Unconventional Environmental Niches*. Retrieved the 5th of May 2016, from http://www.subcultron.eu/.

Thomas, D. (2014). *ROS/Introduction*. Rerieved 31st of March, from http://wiki.ros.org/ROS/Introduction.

Ueland, E. (2016). *Autonomous Guidance and Mapping of Unknown Terrain - By the use of Autonomous Surface Vehicles*. Unpuplished master's thesis, Norwegian University of Science and Technology, Trondheim, Norway.

UNIZG, L. (2016). *aPad #1 - First time in the water*. Retrieved 25th of May 2016, from https://www.youtube.com/watch?v=jY4j4_M4H9s.

US Digital (2015). *US Digital - H6 Datasheet*. Retrieved 17th of November 2015 from http://cdn.usdigital.com/assets/datasheets/H6_datasheet.pdf?k=635833655589557133.

White, J. R. (1997). *Lecture Notes: Introduction to the Design and Simulation of Controlled Systems*. Retrieved 23rd September 2015, from http://www.profjrwhite.com/system_dynamics/sdyn/s7/s7intro/s7intro.html.

Zare, A., Balochian, S. c., Arvan, M., and Balochian, H. (2009). *A new optimal control approach for Double Inverted Pendulum on Cart (DIPC)*. Karachi. Conference of 2009 2nd International Conference on Computer, Control and Communication, IC4 2009 ; Conference Date: 17 February 2009 Through 18 February 2009;.

Zhang, C., Hu, H., Gu, D., and Wang, J. (2016). *Cascaded control for balancing an inverted pendulum on a flying quadrotor*. *Robotica*, pages 1–17. Article in Press.

Appendix A

MATLAB and Simulink Files

All files in this appendix are only included with electronically submitted versions of the thesis.

A.1 Cart-Pendulum run file

run_cartPendNonLin.m

A.2 Cart-Pendulum Simulink Model

cartPendNonLin.slx

A.3 Visualization

Visualization.m

A.4 4 DOF Vessel Model

Saucer_4dof.slx

A.5 4 DOF Vessel Run File

run_Saucer_4dof.m

A.6 MIP Model Without Hydrodynamics

MIPv2.xls

A.7 Runfile MIP Without Hydrodynamics

run_MIPv2.m

A.8 MIP Model With Hydrodynamics

MIPv2_hydro.xls

A.9 Run MIP With Hydrodynamics

run_MIPv2_hydro.m

A.10 Kalman Filter MIP

MIPv2_Kalman.slx

A.11 Kalman Filter MIP Run

run_MIPv2_Kalman.m

A.12 MIP With Hydrodynamics LQR

MIPv2_hydro_LQR.slx

A.13 MIP With Hydrodynamics LQR Run

run_MIPv2_hydro_LQR.m

A.14 MIP LQG

MIPv2_LQG.slx

A.15 Run MIP LQG

run_MIPv2_LQG.m

A.16 Encoder Reading

encoder.slx

A.17 Encoder Reading Run File

run_encoder.m

A.18 Test All Sensors

BattEncThrustRPM.slx

A.19 Arduino Code

- EncoderBatteryThrustVect.ino
- RPMBatteryVect.ino

A.20 Arduino Mega Code

ThrustEncoderBatteryRPM.ino

A.21 Simplified LQR Simulink

PendAndHeadingCtrlLQR.xls

A.22 Simplified LQR Simulink Run File

run_PendAndHeadingLQR.m

A.23 Full State LQR Simulink

MIP_LQR.xls

A.24 Full State LQR Simulink Runfile

run_MIP_LQR.m

A.25 Full State LQG

Saucer_LQG

A.26 Full State LQG Initialization

run_Saucer_LQG

A.27 Launchfile RPi2

Einar.launch

A.28 Own Functions

These are own functions used to simplify calculations related to robotics.

- rad2deg.m - Converts angles from radians to degrees.
- deg2rad.m - Converts angles from degrees to radians.
- rotx.m - Calculates a rotation of alpha degrees about the x axis.
- roty.m - Calculates a rotation of beta degrees about the y axis.
- rotz.m - Calculates a rotation of gamma degrees about the z axis.
- DirectKinematics.m - Calculates the direct kinematics for an industrial robot.
- denhartmat_final.m - Final homogeneous transformation matrix.
- RobotTransform.m - Transformation matrix.
- transform_final.m - Final transformation matrix using the DH convention.

Appendix B

Instruction Manual for the CS Saucer and Pendulum

B.1 About

This manual provides the necessary information needed to install required software, and deploy and run the CS Saucer with the inverted pendulum. It is assumed that the Raspberry Pi 2 used during this project is available. Therefore the manual will not include steps on installing and preparing the Raspberry Pi 2 (RPi2) unit.

- The manual is written and tested using ROS Indigo and MATLAB 2015b.
- Text written in a yellow box should be entered in the Linux terminal window, unless noted otherwise. A terminal can be opened by pressing ctrl + alt + T.
- Gedit is used as text editor in the manual. However, feel free to use your favourite editor.
- The username of the Raspberry Pi 2 unit in the CS Saucer is ubuntu, and the password is mclab123.

B.2 Installing Required Software

- Install Ubuntu on you personal computer. It is recommended to install Ubuntu 14.04.4 LTS. It can be installed via Virutal Box, but Simulink might run slowly. It is recommended to create an own partition.
- Install ROS Indigo in Ubuntu. Follow the steps found here: `http://wiki.ros.org/indigo/Installation/Ubuntu`. It is important that Indigo is selected, as this is the version currently installed on the Raspberry Pi 2.

B.2.1 Getting Started With ROS

First, create your personal workspace by following the steps found here: `http://wiki.ros.org/catkin/Tutorials/create_a_workspace`. Then paste the launch file "Einar.launch" (Appendix A.27) into the workspace you just created (/catkin_ws/src).

To make sure that the setup.bash file is sourced when a terminal window is opened, enter the following:

```
1 echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

To enable communication between your computer and the Raspberry Pi, the ip addresses must match. Open a terminal and enter

```
1 ifconfig
```

This will show the ip address in the terminal window. In this example, the computer have the ip address 192.168.0.102, while the RPi should have the fixed address 192.168.0.232 on the MC Lab network. This must be entered in the hosts file:

```
1 sudo gedit /etc/hosts
```

Enter your password and then add the following to the file that opens in Gedit:

```
1 192.168.0.102 rotem
2 192.168.0.232 ubuntu
```

Save and close the gedit file. The name behind the ip address is the username of the computer. The RPi 2 unit is named ubuntu. The command over must be entered both on your computer, and in the RPi 2. Notice that the ip of the RPi2 already is included in the hosts file in the RPi.

The next step is to export the rosmaster from your personal computer to the RPi 2. This is done by editing the bashrc file:

```
sudo gedit ~/.bashrc
```

Enter your password if prompted, and then add the following line at the bottom of the text document that opens

```
export ROS_MASTER_URI=http://ubuntu:11311
```

Save the document and close.

Now, during testing it might be desirable to experiment through your own computer, without having to work via the RPi. If you want do to this, the line you just entered in the bashrc file must be commented out by entering # in front of it.

You should now be able to source shell both from your computer to the RPi 2, and back. This allows you to work on the RPi OS from your own computer, and the other way around. To enter the RPi from your computer, open a terminal and enter:

```
ssh ubuntu@ubuntu
```

You will need to enter the password of the ubuntu, mclab123. See also step 1 here `http://wiki.ros.org/ROS/NetworkSetup`. You should be able to send ROS messages between the computers. Notice that both have to be connected to the MC Lab network.

B.2.2 Arduino and ROS

The Arduino code required to run the MIP system (Appendix A.20) should already be uploaded to the Arduino Mega embedded circuit board. However, it might be convenient to also have the Arduino IDE and ROS package on your own computer. This is since you might want to change and add/remove elements from the code to fit your application. Note that this can be done via the RPi if you do not want to install Arduino on your computer.

Run the following commands on your computer (this is already done on the RPi2):

```
sudo apt-get install arduino
sudo apt-get install ros-indigo-rosserial
```

You can now open the Arduino IDE as a normal program, and edit and write code. If you did not install Arduino IDE on your computer, you can enter the program from the RPi by doing

```
ssh -X ubuntu@ubuntu
arduino &
```

The -X allows you to open programs from your computer on the remote desktop to which you are accessing through ssh.

B.2.3 Qualisys and ROS

You can attach the Qualisys IR markers to the vessel and use the Qualisys system to determine the position and orientation of the vessel, but first the system must be imported into ROS.

First, import the driver from Github:

```
cd ~/catkin_ws/src
git clone https://github.com/KumarRobotics/qualisys
cd ~/catkin_ws
catkin_make
```

Now open the qualisys.launch file in a text editor and edit the ip address and port number.

```
sudo gedit ~/catkin_ws/src/qualisys/launch/qualisys.launch
```

As of March 2016 the ip should be 192.168.0.10, and the port number is 22222.

To test if you can interface Qualisys and ROS you should try to ping Qualisys from your computer over the MC Lab network. If this is successful, you can "listen" to Qualisys data over ROS.

Put some IR markers in the basin, and make sure they are visible on the Qualisys computer. The CS Saucer body is saved in the system. You should make sure you are in the working directory:

```
cd catkin_ws/src
```

Now run

```
roslaunch qualisys qualisys.launch
rostopic list
```

The first command launches the node, while the last command lists all the topics currently published to ROS. If you have loaded the SaucerRotem on the Qualisys computer, you should see two messages, position and orientation. You can then listen to data by entering

```
rostopic echo EulerAngles
```

This command will print the three Euler angles giving the orientation of the vessel.

Qualisys Data to MATLAB

Most messages sent over the ROS network are standard messages included in the ROS package. The rpm and encoder data are examples of standard messages. The Qualisys system, however, sends a custom message. These type of messages are not automatically recognized by MATLAB (as opposed to standard messages, who are).

The following steps are required for MATLAB to recognize the custom message.

First create a new folder

```
mkdir ~/qualisysDir
```

Now, copy the folder named qualisys, located in /catkin_ws/src, and paste it into the folder qualisysDir you just created.

The next step is to edit the package file so that it is recognized by MATLAB. Open package.xml in a text editor:

```
sudo gedit ~/qualisysDir/qualisys/package.xml
```

Then add the following two lines somewhere in the main body of the package.xml file.

```
<build_depend>geometry_msgs</build_depend>
<build_depend>std_msgs</build_depend>
```

Now you need to download the custom message package into MATLAB. Open MATLAB and enter the following line in the command window:

```
roboticsAddons
```

if you have MATLAB 2016, or

```
roboticsSupportPackages
```

if you are running MATLAB 2015. Follow the instructions on screen to download the ROS custom message package.

When the download is finished, enter the following in the MATLAB command window

```
folderpath= '~/qualisysDir'
rosgenmsg(folderpath)
```

Now follow the instructions given by MATLAB in order generate the needed message type. In this process you may need allow writing permission to the file "pathdef.m"

Start ROS in Matlab by entering

```
rosinit
```

in the command window, or

```
rosinit('ubuntu')
```

if you have exported the rosmaster to the RPi2.

You can now get the data into Simulink by the Subscriber block, or to MATLAB workspace by typing the following commands:

```
Subb = rossubscriber('/qualisys/Saucer');
posedata = receive(Subb,10);
```

Notice that "Saucer" is the name you have given the vessel on the Qualisys computer.

Remember that the Qualisys node always needs to be launched before reading signals in MATLAB.

```
roslaunch qualisys qualisys.launch
```

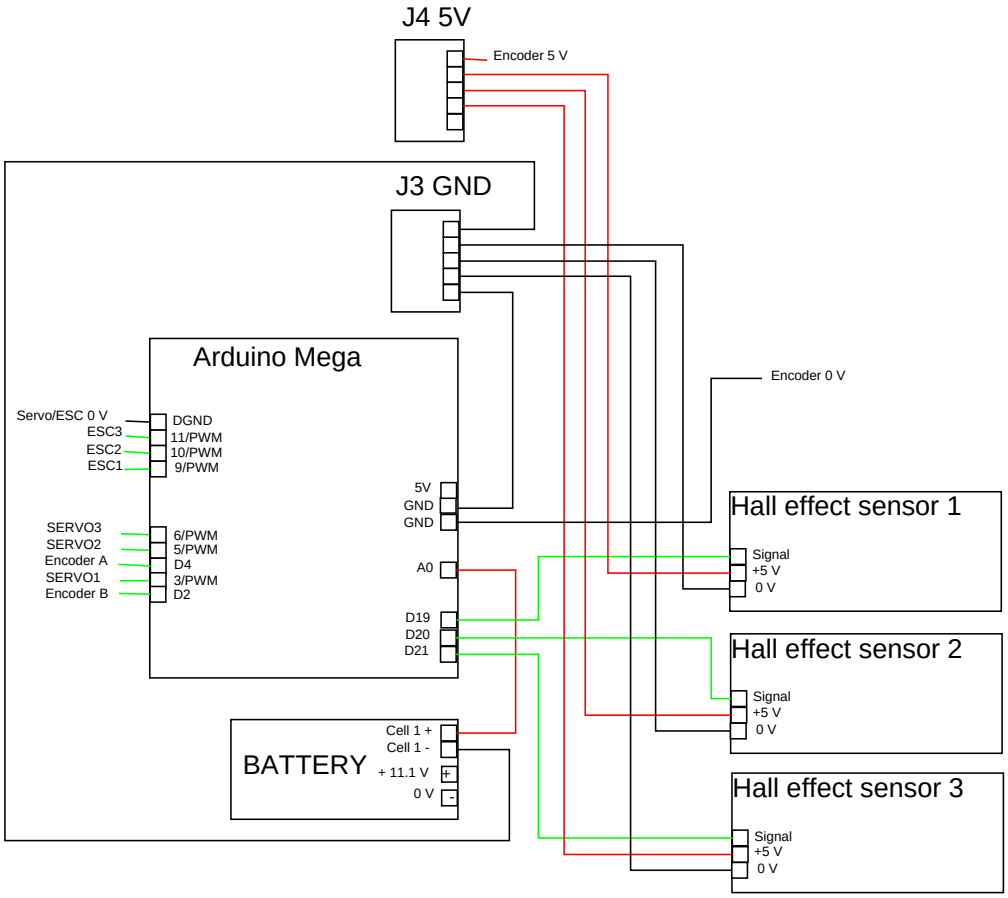


Figure B.1: Wiring diagram to the Arduino mega.

B.3 Launching the CS Saucer and Pendulum

Once all the necessary software have been installed, as described in Section B.2, you are ready to deploy the MIP system for the first time.

First, make sure the battery is fully charged. It should measure about 12.5 V. Then connect the wires according to Figure B.1. Make sure the Arduino is connected to the RPi2 via the USB cable, and that the WiFi dongle is plugged in. Then connect the power from the battery. The three speed controllers should light in constant red green and blue.

Open a terminal window and enter

```
1 ssh ubuntu@ubuntu
```

enter the password for the RPi2: mclab123. Then do

```
cd catkin_ws/src
roslaunch Einar.launch
```

The file Einar.launch launches the Arduino node.

Open MATLAB and enter

```
rosinit('ubuntu')
```

in the command window.

To test the system, you can run the script run_MIP_LQR.m (Appendix A.24). Then open the Simulink diagram in Appendix A.18. Run the Simulink model. You should now see the battery voltage, the encoder value and the three rpm values. You can then enter a value in the manual thruster control, or in the force control to test the thrusters. The rpm measurement should update as the thruster runs.

To run the CS Saucer with pendulum and LQR controller, first make sure the lid is firmly fitted by tightening the four wing nuts. Then open a new terminal window and enter

```
cd catkin_ws/src
roslaunch qualisys qualisys.launch
```

This enables the Qualisys node in ROS. Notice that the Qualisys markers must be visible on the Qualisys computer prior to running the node.

Notice that if you experience problems, you might need to restart ROS, or at least restart ROS in MATLAB. Do this by entering

```
rosshutdown
rosinit('ubuntu')
```

in the command window.

Now run the script run_MIP_LQR.m (Appendix A.24). Open the Simulink file MIP_LQR.slx (Appendix A.23), and run it. The vessel will now try to balance the pendulum using a LQR controller. The heading is also controlled with a PD regulator. Notice that there is a manual switch in the diagram to turn on and off all the thrusters.

Appendix C

Parameters

Gain for the LQR:

$$K_{lqr} = \begin{bmatrix} 0.1000 & -8.4703 & -240.8594 & 1.3717 & -8.8663 & -108.0853 \end{bmatrix} \quad \text{(C.1)}$$

Gain for the Luenberger Observer. System noise is applied.:

$$L = \begin{bmatrix} 43.9917 & -5.8391 & 6.9748 \\ 4.6831 & 43.5908 & -2.8069 \\ -3.1679 & 2.5961 & 44.2645 \\ 500.3468 & -125.8514 & 170.7909 \\ 117.7478 & 512.8949 & -54.4017 \\ -75.8818 & 57.5201 & 517.7372 \end{bmatrix} \quad \text{(C.2)}$$

Table C.1: Parameters used for the marine inverted pendulum simulations.

Parameter	Name	Value
Gravity	g	9.81 $\frac{\mathrm{m}}{\mathrm{s}^2}$
Water density	ρ	1000 $\frac{\mathrm{kg}}{\mathrm{m}^3}$
Vessel mass	m_0	6.34 kg
Vessel radius	r	0.274 m
Vessel height	h	0.129 m
Vessel mass moment of inertia	$I_y = I_x$	0.128 kgm^2
Vessel mass moment of inertia	I_z	0.116 kgm^2
Added mass	$X_{\dot{u}}$	-3.5 kg
Added mass	$Y_{\dot{v}}$	-3,5 kg
Added mass	$M_{\dot{q}}$	-3 kgm^2
Linear damping	$X_u = Y_v$	-1.96 $\frac{\mathrm{kg}}{\mathrm{s}}$
Linear damping	M_q	-3 $\frac{\mathrm{kgm}}{\mathrm{s}}$
Nonlinear damping	$X_{\lvert u\rvert u} = Y_{\lvert v\rvert v}$	-7.095 $\frac{\mathrm{kg}}{\mathrm{s}}$
Nonlinear damping	$M_{\lvert q\rvert q}$	-12 $\frac{\mathrm{kgm}}{\mathrm{s}}$
Metacentric height	$\overline{GM}_L$	0.653 m
Arm	l_0	0.15 m
Arm	r_t	.1 m
Pendulum length	l_1	1.5 m
Pendulum mass	m_1	0.2 kg
Pendulum inertia	I_1	0.15 kgm^2

*Parameters not listed are zero, if used.