

Group 1034



IAS Master Thesis

MPC and LQ Control of an Autonomous Helicopter
Dragging a Semi-submerged Slung Load

Steffen Vutborg
Stefan Moeskjær Pedersen

Aalborg University 2012

Front page picture is a modified version of a picture taken from <http://coveroid.com/> showing a Search and Rescue Helicopter over water.

Synopsis:

Title:

MPC and LQ Control of an
Autonomous Helicopter Dragging a
Semi-submerged Slung Load

Theme:

Intelligent Autonomous Systems

Project term:

IAS Master Thesis, 2012

Project group:

12gr1034

Group members:

Stefan Moeskjær Pedersen
Steffen Vutborg

Supervisors:

Anders la Cour-Harbo
Morten Bisgaard

Number of copies: 5

Number of pages: 188

Number of appendices: 3

Completed: 2012.05.31

After investigating whether an autonomous helicopter is a viable search and rescue opportunity, including a correspondence with a search and rescue helicopter pilot, the project is delimited to a proof of concept system with a helicopter dragging a semi-submerged slung load.

Using a finished helicopter model referred to as the Bisgaard model, the modelling part includes a model of a semi-submerged slung load, as well as a rigid body model describing the interactions between the helicopter and the slung load. Though, with the wire attachment point in the helicopter centre of mass. Experimentation showed the slung load model to have slightly different dynamics, and not enough damping when compared.

The control part of this work includes a series of LQ controllers for the system both with and without slung load attached, with an expansion of an outer loop MPC to control the set-points. The part also include switching between the controllers, making the system able to fly to a reference position and change to hover.

The results from the acceptance test show how the produced controllers are able to stabilise the system to a degree where the requirements are fulfilled, but with plenty of control opportunities open for future work.

Preface

This report is produced by group 1034 on third and fourth semester of the Intelligent Autonomous Systems master program at Department of Electronic Systems, Section for Automation & Control at Aalborg University. The purpose of these semesters are, according to the study program:

"Using the knowledge in modelling and control from the previous semester the focus is on introducing autonomy into a mechanical or energy conversion system. This includes the use of estimation methods for determining the (sub)optimal system parameters for the purpose of increase self-containment of the system."

To fulfil this purpose of the study program, it is chosen to make an autonomous naval search and rescue helicopter. Literary references used in the project are denoted by number [#], like such: [5]. Figures, tables and equations are denoted "figure #.#", "table #.#" and "(#.#)" respectively, referring to the chapter of the figure, table or equation, and the number of that figure, table or equation in this chapter. The bibliography is listed in the back of the appendix.

A digital copy of the report, MATLAB scripts and simulation data are on the enclosed DVD. The DVD is located at the back of the report, and when referred to, it is shown with a picture followed by the directory path, as such:



[DIRECTORY]

Acknowledgement

A great thanks goes to Matthieu Verdy, who worked on this project from September 2011 to December the same year, as partial fulfilment of his electrical engineering education. His work with the project and specifically within the field of hydrodynamics is highly valuable to the results.

Author Signatures

Stefan Moeskjær Pedersen

Steffen Vutborg

Contents

Preface	iii
Contents	iv
Introduction and Motivation	xi
I Problem analysis	1
1 Previous Work	3
1.1 Helicopter and Slung Load Modeling	3
1.2 Submerged Load Modelling	4
2 Helicopter Basics and Terminology	5
2.1 Introduction to Helicopter Mechanics	5
3 Description of the Case System	9
3.1 Helicopter Description	9
3.2 Slung Load Description	10
4 Problem and Requirements	13
4.1 The Problem	13
4.2 Requirement Specification	15
4.3 Acceptance Test Specification	16
II Model	21
5 Helicopter Model	23
5.1 Coordinate Systems	24
5.2 Forces and Torques	25
5.3 Helisim	27
5.4 Helicopter model conclusion	28
6 Semi-Submerged Load Dynamics	29
6.1 Hydrostatic Effects	30
6.2 Added Mass	31
6.3 Damping Effects	33
6.4 Lift and Drag Force Corrections and Simulations	34
6.5 Semi-submerged Load Conclusion	44

7	Rigid Body Model	45
7.1	Degrees of Freedom and State Variables	46
7.2	Conservation of Momentum	49
7.3	Equation Summary	53
8	Model Simulation and Verification	55
8.1	Slung Load Simulation	55
8.2	Vertical Movement Experiment	56
8.3	External Force Simulations	58
8.4	Conclusion of Slung Load Model Simulations	59
8.5	Rigid Body Model Simulations	60
9	Model Implementation for Helisim	69
9.1	Semi-submerged Load Dynamics Implementation	70
9.2	Rigid Body Model Implementation	71
	III Controller	73
10	General Control Strategies	75
10.1	LQ Control Considerations	76
10.2	Model Predictive Control Considerations	83
11	Linear Quadratic Control	85
11.1	LQ Control Methodology	85
11.2	LQ Control of a Standalone Helicopter	87
11.3	LQ Control of a Helicopter with Slung Load	97
11.4	LQ Controller Stability Regions	108
11.5	LQ Control Conclusion	110
12	Model Predictive Control	113
12.1	Model Predictive Control Methodology	113
12.2	Model Predictive Control of a Standalone Helicopter	121
12.3	Model Predictive Control of a Helicopter with Slung Load	129
12.4	Model Predictive Control Conclusion	136
	IV Evaluation	139
13	Acceptance Test	141
13.1	Standalone helicopter acceptance tests	142
13.2	Full system acceptance tests	151
13.3	Acceptance test conclusion	161
14	Comparison of the Controllers	163
14.1	Standalone Helicopter Controller Comparison	163
14.2	Helicopter with Slung Load Controller Comparison	165
15	Conclusion	169

V Appendices	173
A Projected reference area	175
B Linearisation of Helicopter and Slung Load Model	179
B.1 Standalone Helicopter Model Linearisation	179
B.2 Helicopter with Semi-submerged Slung Load Model Linearisation	180
C Moving and Rotating Noninertial Frames	181
C.1 Moving frames	181
C.2 Moving and rotating frames	182
C.3 Example: Rotation and velocity difference	184
Bibliography	187

Nomenclature

Abbreviations

COM	Centre Of Mass, page 24
DOF	Degrees of Freedom, page 6
GPS	Global Positioning System, page 9
IMU	Inertial Measurement Unit, page 9
LQ	Linear Quadratic, page 85
MPC	Model Predictive Control, page 113
RHC	Receding Horizon Control, page 113
SAR	Search and Rescue, page xi
UAS	Unmanned Aerial System, page xiii
UAV Lab	Unmanned Aerial Vehicle laboratory, page 3

Symbols

α	Angle of attack, page 34
$\Delta \hat{\mathbf{u}}$	Predicted change in control signal, page 116
$\dot{\mathbf{q}}_b$	Velocity vector of the full system, page 47
$\hat{\mathbf{u}}$	Predicted control signal, page 116
$\hat{\mathbf{z}}$	Predicted performance signals/State values, page 116
\mathbb{E}	The earth fixed frame, page 24
\mathbb{H}	The helicopter centre of mass fixed frame, page 24
\mathbb{L}	The load centre of mass fixed frame, page 24
\mathbf{A}	State space system matrix, page 75
\mathbf{B}	State space input matrix, page 75
\mathbf{C}	State space output matrix, page 75

I	Moment of inertia matrix, page 49
K	Control matrix, page 75
Q	State weight matrix, page 86
R	Input weight matrix, page 86
ω_n	Pole natural frequency, page 79
${}^{\mathbb{P}}\phi$	<i>Pitch</i> : Tilting forwards and backwards around the y-axis in frame \mathbb{P} , page 6
${}^{\mathbb{P}}\psi$	<i>Yaw</i> : Rotating left and right around the z-axis in frame \mathbb{P} , page 6
${}^{\mathbb{P}}\theta$	<i>Roll</i> : Tilting side to side around the x-axis in frame \mathbb{P} , page 6
${}^{\mathbb{P}}x$	First position coordinate, expressed in frame \mathbb{P} , page 6
${}^{\mathbb{P}}y$	Second position coordinate, expressed in frame \mathbb{P} , page 6
${}^{\mathbb{P}}z$	Third position coordinate, expressed in frame \mathbb{P} , page 6
${}^{\mathbb{Q}}_{\mathbb{P}}\mathbf{R}$	Rotation matrix expressing the rotation from \mathbb{P} to \mathbb{Q} , page 48
ρ	Density, page 33
χ	State vector, page 47
ϵ	Error between set point trajectory and measurement, page 114
ω_h	Angular velocity of the helicopter, page 47
ω_l	Angular velocity of the load, page 47
τ	Torque vector, page 49
\mathbf{a}_h	Linear acceleration of the helicopter., page 50
\mathbf{F}	Force vector., page 49
\mathbf{L}_l	Angular momentum of the load, page 51
\mathbf{q}	Position vector of the full system, page 46
\mathbf{r}	Reference trajectory, page 114
\mathbf{s}	Set-point trajectory, page 114
\mathbf{v}_h	Linear velocity of the helicopter, page 47
ζ	Pole damping ratio, page 79
${}^{\mathbb{E}}\boldsymbol{\theta}_h$	Euler angles of the helicopter, expressed in the \mathbb{E} frame, page 46
${}^{\mathbb{E}}\boldsymbol{\theta}_l$	Euler angles of the load, expressed in the \mathbb{E} frame, page 46
${}^{\mathbb{E}}\mathbf{p}_h$	Position of the helicopter, expressed in the \mathbb{E} frame, page 46

C_d	Drag coefficient, page 33
C_l	Lift coefficient, page 34
g	Gravitational acceleration, page 30
H_p	Prediction horizon, page 116
H_u	Control horizon, page 116
k	Current discrete time step, page 113
m	Mass, page 30

Introduction and Motivation

Accidents at sea has always been a cause for loss of lives and one of the first well documented Search and Rescue (SAR) missions was conducted in 1656 in search for the Dutch East India Company ship Vergulde Draeck. It wrecked on the Australian west coast, near Ledge Point on April 28th. A search party of two ships were dispatched in the SAR mission. The rescue mission was unsuccessful, and one of the two rescuing ships was lost in the mission [17].

In more recent history, helicopters are often used due to the speed, versatility and overview they provide. The first naval rescue by helicopter was in 1947 where Sikorsky's chief test pilot Dimitry Viner was on an aircraft carrier to demonstrate the capabilities of Sikorsky's new S-51 helicopter to the navy. During the demonstration lieutenant Robert Shields had to crash-land his SB2C bomber in the ocean. Viner heard of lieutenant Shields' accident, and flew to his aid. Under ten minutes after his mayday, lieutenant Shields reported in for duty on the aircraft carrier [4]. Since then, helicopters have been used in countless SAR missions.

The first helicopter SAR fleet appeared in Denmark in 1957, as a response to frequent fighter plane crashes. It was composed of seven Sikorsky S-55 helicopters, which greatest disadvantage was having a relative short operational range. Due to the short range of the S-55, another helicopter was sent out searching for the distressed persons, and the S-55 was sent directly to the site.

Today Denmark uses the AgustaWestland EH101 helicopter for SAR missions. A single of these helicopters have an stock unit price of 21 million US\$, and besides from that also has a high maintenance and running cost. A smaller autonomous solution could aside from having a faster response, also be cheaper than the current non-autonomous SAR solutions.

Statistics on casualties in the Danish waters are not available, but according to [14] a total of 42.133 lives have been lost at sea in the period from 1970 to 2005, within the US Coast Guard's domain alone. At least 66% of these deaths occurred after the coast guard had been notified. Some of these lives might have been saved with a faster response time.

Interview with a Search and Rescue Helicopter Pilot

To get a better insight into how SAR missions are conducted, and other background information, contact is made to a SAR helicopter pilot Thomas Wandahl who answered a series of questions in a mail correspondence. The answers from Thomas Wandahl is given using his own opinions, and is not a reflection of the opinions of the navy or any other organisations. Answers are only estimates, and is answered to be the best of his knowledge.

In a question about the costs and personnel needed for a SAR mission, Thomas Wandahl answered that it vary a lot depending on the nature of the mission. Regarding costs of a mission, the price of running the rescue helicopter used in Denmark, which is the EH101, is about 60 000^{DDK}/hour plus the salary of everyone involved. The amount of personnel needed vary from around 5 persons when the mission is an evacuation of an injured sailor, up to more than 20 people from rescue agencies in a SAR mission where the exact location of the distressed person is unknown.

In another question regarding the response time from distress call to take-off, his reply is that the average response time from 7am to 10pm is around 10 minutes, and around 20 minutes the rest of the time. An estimate on the time from distress call to the helicopter arriving at the site is below 30 minutes in most cases, the time is however increased significantly if the current location of the distressed person is unknown. Meaning that if the time to arrival can be decreased, and the search for the person in distress could be automated, the total time from distress call to actual rescue could be lowered.

To a question about the size and load capacity of the EH101, he replies that it definitely do not have to be that large for simpler SAR missions, which also is why some other countries uses smaller helicopters such as Sea King, Sea Hawk and Super Puma. A significant advantage having a SAR helicopter as large as the EH101 is the on board treatment facilities, that smaller manned or autonomous helicopter can not accommodate. The capacity is also an advantage if the rescue mission involves multiple persons, Thomas Wandahl does however estimate that this only covers 5% of missions, with the remaining 95% being missions involving single persons in distress, which is a plus for a smaller autonomous SAR system, only capable of single person rescue missions.

Mr. Wandahl's answers goes on to questions regarding a semi-submerged slung load being dragged through the water, where he comment that if the slung load is a person, it is not recommended, since humans suffers damages easily at higher velocities, meaning that the velocity must be kept low to keep the person from taking damage. He also comments that an autonomous helicopter system like the one in this work, needs the ability to release the load if something goes wrong.

Technological Motivation

Another motivating aspect of the problem is from a research point of view, where an Unmanned Aerial System (UAS) carrying/dragging a partially submerged slung load, is still to be seen. This feature is an interesting modification of the already existing models of helicopter with slung-load systems, expanded to include hydrodynamic effects on the slung load and how these effects translate to the helicopter.

Since most modelling of hydrodynamic forces assumes the object being modelled to be totally submerged in a fluid, it is relevant for this work to investigate what happens when only part of the slung load is submerged, but more importantly what happens to the forces when the submersion depth changes, and how could this depth change be included in the computations of the forces.

With the changing depth of the semi-submerged slung load included into the computations of the hydrodynamic forces, another interesting research subject, from a modelling point of view would be how the newly included effects translates to the helicopter, through the modelling of two connected rigid bodies.

The application options for an autonomous helicopter system with a semi-submerged slung load could include search and rescue missions of distressed persons in water as in this work, but also dragging of smaller boats. Another application option could be material transport at the construction of floating or seabed mounted structures like oil rigs or floating wind turbines. In larger scale, or with a series of helicopters could applications also include relocation of larger structures.

Part I

Problem analysis

Part introduction

Contents

- 1 Previous Work
- 2 Helicopter Basics and Terminology
- 3 Description of the Case System
- 4 Problem and Requirements

Purpose

This part titled Problem Analysis start out by covering some of the work already done within the field of autonomous helicopters and slung loads

The part also covers a chapter introducing the terminology and basics of small scale helicopters. A chapter about the specific helicopters and slung load used in this work has been added as well, as an extension to the helicopter introduction, this is done so any references to the helicopter can be used with its name, and so the specifications can be used in the modelling and control parts of the report.

The final chapter of the problem analysis section covers the requirement specification including the delimitation of the problem and the acceptance test specifications.

Goals

- *Introduce relevant work done within the field of autonomous helicopters using slung loads.*
- *Introduce the basic mechanics and terminology of small scale helicopters.*
- *Describe the specifications of the helicopter and slung load used in this work.*
- *Formulate a requirement specification including problem delimitation.*
- *Define the acceptance test specification.*

Previous Work

This chapter will present research concerning helicopter modelling and control, with slung load. Several projects and articles has been written about these topics, and the Unmanned Aerial Vehicle laboratory (UAV Lab) of Aalborg University has been researching and working with the topic for a number of years.

1.1 Helicopter and Slung Load Modeling

Markus Bernard and Konstantin Kondak [5] published a paper in 2009 concerning autonomous slung load transportation using multiple small scale helicopters. The modelling is done using Kane's method. Initially the masses and dynamics of the suspension lines for the slung load was omitted, which led to oscillations in the lines that propagated throughout the system. To solve this Bernard and Kondak introduced a state observer for the slung load position. After discovering this problem, they created a robust controller, that was able to control a constellation of helicopters, both in simulation and in real flight.

R.A. Stuckey [19] wrote an article in 2001 concerning the stability and control problems in helicopters and slung load systems. This work has led to the creation of an algorithm able to successfully model a helicopter with slung load. The model contains both elastic and inelastic formulations concerning links between helicopters and slung loads.

In 2007 Morten Bisgaard [6] wrote a PHD thesis concerning modelling and controlling of an autonomous helicopter with slung loads. This thesis presents a model of a helicopter with a slung load system attached to be used for generic cargo helicopter and land mine detection. The thesis explains the modelling of the helicopter and slung load systems. It also contain chapters regarding estimation and design of a controller for such a system. Theoretical results was consistent with practice tests and analysis.

The helicopter modelling in Bisgaard's work was done by first analysing the physical dynamics of the interesting parts of the helicopter (body, blades, rotors, etc.), which have then been described in equations. Then, the model of the slung load is added to get a complete model of the system.

D. Fusato and G. Guglieri [10] published an article in 2001, concerning flight dynamics of an articulated rotor helicopter with an external slung load, where the main focus of the article was on the rigid body modelling, which in this case means the interactions between the helicopter and slung load. They obtained results for slung loads with up to 28% load-to-helicopter mass ratio, where they found out that the influence of cable lengths in the range of 3-8 meters was negligible for a real heavy duty helicopter.

1.2 Submerged Load Modelling

Dr. Gianluca Antonelli published a book [2] describing the model and the control of Unmanned Underwater Vehicle (UUV).

Asgeir J. Srensen of the Norwegian University of Science and Technology published a set of lecture notes[20] in 2011 concerning advanced modelling control of motion and propulsion of Ships and ocean structures, with special attention to the effect that wind and waves have on these structures.

Helicopter Basics and Terminology

This chapter will include an introduction into the basics of a helicopter, of the type used in this work. To make the understanding of the different mechanics used in the remaining parts of this work easier. Throughout this chapter, different terms and notations are introduced, that will be used in the remaining parts of the report.

2.1 Introduction to Helicopter Mechanics

First an explanation of the helicopter, and the basics of how the steering mechanisms work is wanted. However, a short introduction into helicopter terminology is needed to proceed.

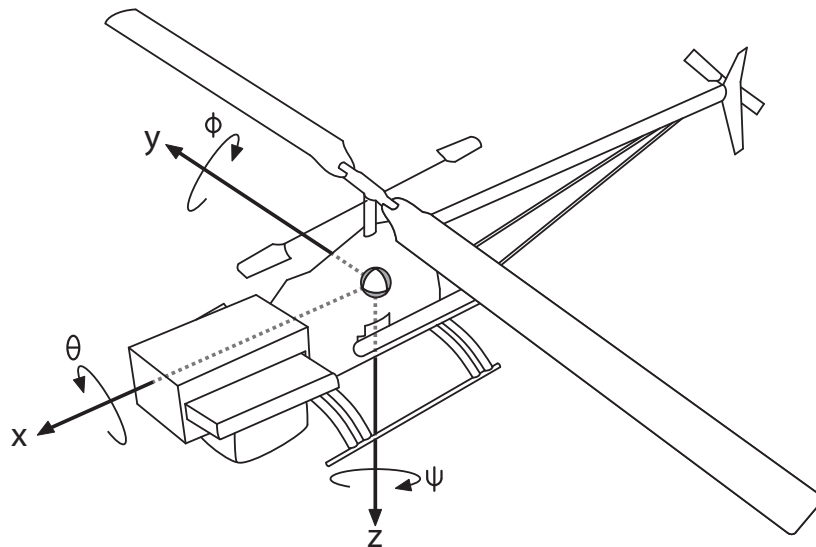


Figure 2.1: An illustration of a helicopter with a body fixed frame of reference inserted showing the heading of the helicopter.

Figure 2.1 on the preceding page show the heading of the helicopter by describing the angles using the terms *Pitch*, *Roll* and *Yaw*, also called the Euler angles as described below. When addressing the heading of the helicopter in the remaining parts of this work these angles are used.

ϕ	<i>Roll</i> : Tilting side to side around the x-axis.	rad
θ	<i>Pitch</i> : Tilting forwards and backwards around the y-axis.	rad
ψ	<i>Yaw</i> : Rotating left and right around the z-axis.	rad

While the pitch roll and yaw angles describe the heading of the helicopter, the position is described using the x-, y- and z-position components, and not *Heave*, *Sway* and *Surge*.

x	Moving forward and backward along the x-axis.	m
y	Moving left and right along the y-axis.	m
z	Moving up and down along the z-axis.	m

The heading and position of the helicopter adds up to a total of 6 Degrees of Freedom (DOF) whose rotation and movement is to be modelled and later controlled.

2.1.1 The Main Rotor

The main rotor is the source of a helicopter's lift, which is needed to sustain flight. Asides from providing the needed lift, the main rotor is also the part of a helicopter that control the pitch and roll rotations, resulting in changes in the surge and sway velocities.

The structure of the main rotor is shown in figure 2.2. Subsequently it will be explained how difference in lift is generated by changing the *collective blade pitch*, along with how pitch and roll rotations is achieved by generating a *cyclic blade pitch* though the *swashplate*.

One of the essential parts of the main rotor is the swashplate. As illustrated in figure 2.2, the swashplate is controlled by 3 actuators, controlling the position and tilt of the plate.

The position of the swashplate controls the pitch of the blades. This is known as *collective blade pitch* and translates to the angle of the blades. The collective blade pitch control the amount of lift the main rotor generates.

The orientation and angle of the swashplate control the *cyclic blade pitch* meaning the blade pitch of each blade changes over a cycle of one rotation. The orientation of the plate dictate the phace, and the angle dictate the magnitude of the blade pitch changes.

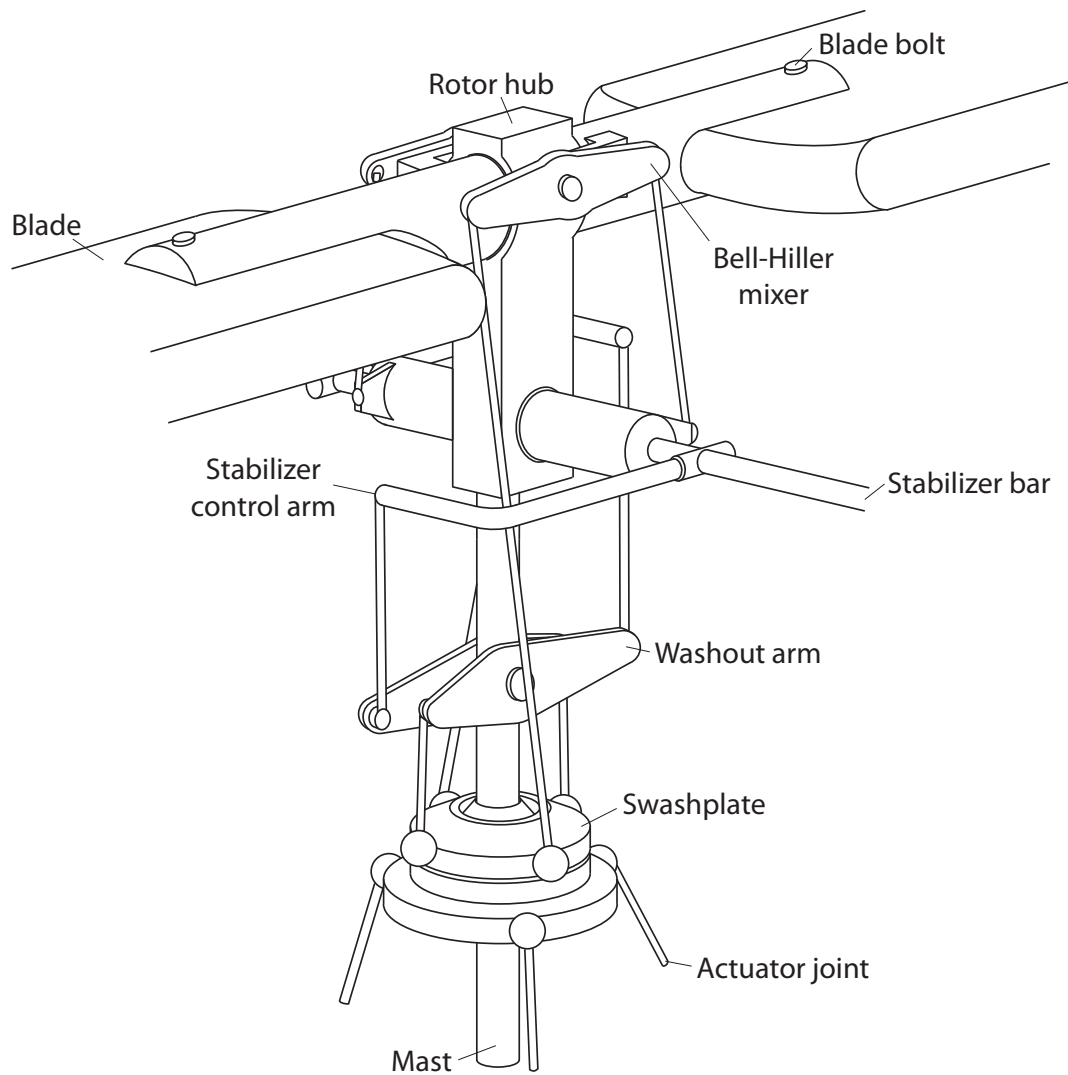


Figure 2.2: An illustration of the main rotor system of a helicopter with a stabilizer bar, showing how the swashplate control collective and cyclic blade pitch.

2.1.2 The Tail Rotor

The tail rotor is the part of a helicopter that control the yaw motion, as well as canceling out the torque applied on the helicopter body from the main rotor. An illustration of the tail rotor mechanics is shown in figure 2.3.

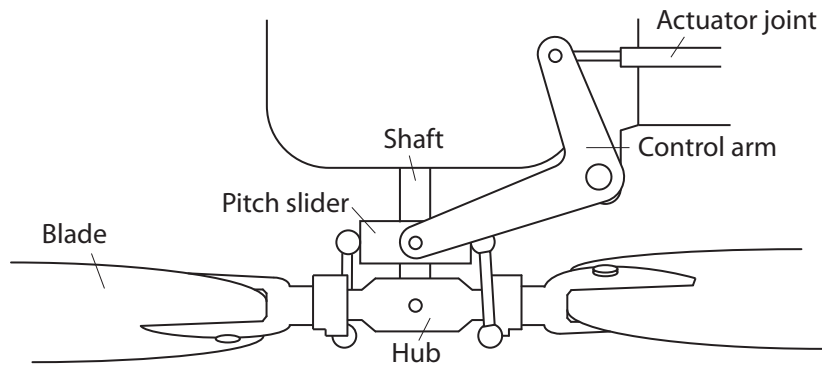


Figure 2.3: An illustration of the tail rotor seen from below, showing how the the collective blade pitch of the tail rotor is changed.

The tail rotor is driven by the same motor that powers the main rotor, so there is no means of changing the angular velocity of the tail rotor directly, since it is directly proportional to the angular velocity of the main rotor with a gearing ratio. Instead the collective blade pitch of the tail rotor is used to control the yaw of the helicopter.

Description of the Case System

This chapter is partitioned into two sections: One describing the helicopter used in this work, and one describing the slung load which the helicopter is going to drag.

3.1 Helicopter Description

The helicopter model presented in part II on page 22 is the model derived in [7]. With the helicopter explored being the same one as the one used in this project, except from some alterations to the on board hardware. This helicopter is based on the Bergen Industrial Twin helicopter with a 52ccm Zenoah 2-cylinder gasoline engine, providing 8 bhp. It weighs about 13 kg and is approved for a total take-off mass of 19 kg, leaving approximately 6 kg of payload. The stock helicopter is capable of approximately half an hour of flight, which is reduced by the increased mass of the helicopter. From this point on, when a helicopter reference is made, the helicopter in question is the AAU-Bergen helicopter. Some of the alterations that makes the helicopter differ from the standard one, is that the following sensors have been fitted:

- **Global Positioning System (GPS)** for position measurements.
- **Magnetometer** for measuring the strength and direction of the surrounding magnetic field.
- **Inertial Measurement Unit (IMU)** for measuring acceleration and angular acceleration.
- **Camera** for visual data recording, such as the location of a slung load.

A control unit in the form of a Acer laptop is fitted in front of the helicopter, as it can be seen in figure 3.1. Underneath the helicopter brackets are fitted, for the attachment of wires for the slung load. This particular helicopter is not suited for real SAR missions, as it can neither carry nor drag a person to shore, but it will suffice for a proof of concept system.



Figure 3.1: The modified AAU-Bergen Industrial Twin Helicopter.

3.1.1 Computer Specifications

The on-board Acer laptop run a standard Linux installation. Real-time computing is simulated by MATLAB/Simulink. C/C++ code can be used interfaced by MATLAB S-Functions. Due to large vibrations, the hard-disk has been replaced by a CF card. The use of a laptop in this kind of environment is unconventional, but electronic components and power supplies can interfere with the radio receiver, and a laptop offer good shielding of these components [13].

3.2 Slung Load Description

No previous work has been found dealing with the towing of a semi-submerged load by a helicopter, the choice for a load has been made based on the fact that it has to simulate a person being pulled. The load is chosen to be a spherical object with a cable attachment point at the top. For this a polyform A-series buoy is used. A picture of such a buoy is shown in figure 3.2.

The usual purpose of such a buoy is to provide a high amount of buoyancy, which is achieved by inflating the buoy with air, giving it a density much lower than that of the surrounding water. The average human density is 1010kg/m^3 [3], which is close to the density of water. This can also be seen as most humans are barely able to float, even with their lungs full of air. To accommodate for this the buoy will be filled with



Figure 3.2: Picture of a polyform A-series mooring buoy.

water, until a suitable density is reached. Though having a comparable density to that of a real person, adding water will enable the centre of mass within the slung load, to move around. The aim density is that of an ordinary person, with a life jacket on. This case assumes a 80 kg person with a life jacket, providing a lift of 150 N, yielding an equivalent density of 809.06 kg/m^3 . Since the helicopter, described in section 3.1 needs to be able to lift the buoy, the smallest one is chosen. The physical parameters of the polyform A-0 mooring buoy are shown in table 3.1. The amount of water that needs to

Provided parameters		Derived parameters	
Diameter	$228.60 \cdot 10^{-3} \text{ m}$	Density	101.52 kg/m^3
Length	$292.10 \cdot 10^{-3} \text{ m}$	Added water	$4.40 \cdot 10^{-3} \text{ m}^3$
Tare mass	$635.02 \cdot 10^{-3} \text{ kg}$	Gross mass	5.06 kg
Buoyancy	5.90 kg		

Table 3.1: Physical parameters of the polyform A-0 buoy. The density and added water, gross mass are derived from the diameter and weight, assuming a perfect sphere.

be added in order to compensate the density is calculated based on these parameters. According to these calculations, a total of $4.40 \cdot 10^{-3} \text{ m}^3$ of water needs to be added, to attain a density of 809.06 kg/m^3 . Adding this amount of water will bring the total mass of the slung load up to 5.06 kg, which is just within the lifting capabilities of the AAU-Bergen helicopter. This corresponds well to a real case, where an autonomous rescue helicopter would be constructed as small as possible, while still being able to drag a person.

Problem and Requirements

The first topic in this chapter is the problem statement and delimitation of the project, and afterwards the requirement specification, ending up in the description of a series of acceptance tests designed to investigate whether the set requirements are fulfilled.

4.1 The Problem

The iterations from the initial problem statement, going through delimitation, and ending up with a final problem statement are examined in this section.

Initial problem statement

How can a SAR UAS, that is able to locate and tow a distressed person at sea be modelled and controlled?

4.1.1 Initial Scenario Description

A description of the scenario that a SAR UAS would have to operate within is needed to find the main focus of this work. The rescue helicopter should be capable of doing the following steps by itself:

1. **Fly to a distressed person:** Upon receiving a distress signal, containing the GPS coordinates of a distressed person, it will take off and fly to the location.
2. **Search of distressed person:** Once at the specific coordinates, it will search for the person using thermal imaging or the like.
3. **Attachment of distressed person:** After the localisation of the person, the helicopter will hover above the person, while lowering down a life jacket or mounting equipment, and wait for the person to become attached to the wire.
4. **Towing of distressed person:** With the person wearing the life jacket, the helicopter should head back for the nearest shore, while towing the person.

The 4 steps mentioned above, all have to be fulfilled for a SAR UAS to be successful, given that the particular helicopter is unable to lift the person completely out of the water.

4.1.2 Problem Delimitation

Constructing a SAR UAS capable of performing the scenario described in the above section would be much too costly and time consuming, given the confines of this work. Therefore a simplification of the task has to be made.

Since the helicopter available for this project, described in section 3.1, is not nearly large enough to carry, nor drag a person, the slung load is downsized. The slung load should however still have approximately the same density as a person wearing a life vest, for more realistic results. The slung load is, for simplicity in the modelling part, seen as a spherical object. While the product of this work will not be a real SAR vehicle, there will be no radio-, or satellite transmissions to the rescue service.

To simplify the overall task, the only parts of the scenario that will be dealt with is the following: Part **1**, assuming that the helicopter start in a hover position, and is given a reference position. Part **3** simplified to hover above the reference position. Part **4**, assuming the helicopter with slung load starts in a hover position, and then start towing the slung load to a new reference position.

Take off and landing is not to be part of the control, as well as the actual attachment process of the slung load should not be either. The search of the distressed person part of the scenario is not treated in this work either, since this work will focus on modelling and control rather than image processing.

4.1.3 Final Scenario Description

Given the above delimitation, the scenario is now that the helicopter, in cases both with and without slung load, should be positioned in hover while in Remote Controlled (RC) mode. Before enabling the autonomous controller. And should be switched back to RC when the helicopter is done dragging the slung load or fly to a reference position, and is hovering.

Final problem statement

How can a UAS that is capable of autonomously towing a semi-submerged slung load through water, be modelled and controlled?

4.2 Requirement Specification

Before the modelling of the system can begin, a set of requirements to the system have to be specified, so the produced model can be used for the control wanted in this work. To find out whether the requirements have been fulfilled, a series of tests is presented in section 4.3 and later conducted in chapter 13 on page 141.

The requirements have been partitioned into two sections. One section regarding the standalone helicopter, and one regarding the full system i.e. the helicopter with a semi-submerged slung load attached. These requirements are denoted by "H" and "S" respectively. These two groups are then split into primary and secondary requirements. Primary requirements have to be fulfilled for this work to be considered a success. Secondary requirements are not as important as the primary ones, but should still be included in the model and control parts of this work. Primary requirements are denoted by a "P" followed by its number e.g. **HP1** denotes the first primary requirement for the standalone helicopter.

Standalone helicopter requirements	
No.	Requirement specification
HP1	Hovering of the helicopter, in a steady position above a certain point.
	The helicopter should be able to hover steadily above the reference position, to act as if a person is being attached as a slung load. The helicopter should be able to hover, within a distance of 0.50 meters horizontally and 0.50 meters vertically.
HP2	Flying towards a reference position.
	The helicopter will need to be able to fly from one location to another in order to perform its tasks, therefore it should be able to fly from its current location to one specified by a reference position, with a flight velocity towards the destination within 1.00m/s of the specified velocity, not counting startup and slowdown. It should be able to go from hover to flight in less than 10.00 seconds without deviating more than 0.50 meters in altitude.
HS3	Arriving at a reference position.
	The helicopter should upon arrival at a reference position be able to go from flight to hover, at the given position in less than 10.00 seconds, while maintaining a maximum deviation in altitude of 0.50 meters. When returned to hover the helicopter should be able to stay, within a distance of 0.50 meters horizontally and 0.50 meters vertically of the reference position.

Table 4.1: Requirements for the autonomous helicopter with no slung load attached.

Full system requirements	
No.	Requirement specification
SP1	Hovering of the helicopter, in a steady position above the semi-submerged load.
	The helicopter should be able to hover, above the semi-submerged slung load within a distance of 0.50 meters horizontally and 0.50 meters vertically.
SP2	Flying towards a reference position.
	The helicopter will need to be able to tow the distressed person from one location to another, therefore it should be able to fly from its current location to one specified by a reference position, while dragging the semi-submerged slung load. The helicopter should be able to transcend from hover to flight condition in less than 10.00 seconds, while not deviating more than 1.00 meter in altitude. While in flight it should be able to maintain a velocity towards the reference position with a maximum deviation of 1.00m/s.
SS3	Preventing fully-submerged load, while in flight.
	While the helicopter is dragging the the slung load towards a given reference position, it should at all time generate enough lift to prevent the slung load from becoming fully-submerged. This is to imitate a person being dragged, while keeping it from drowning in the rescue process.
SS4	Arriving at a reference position.
	The helicopter with slung load should upon arrival at a reference position be able to go from flight to hover, at the given position. It should do so in less than 10.00 seconds without deviating more than 0.50 meters in altitude. When in hover, the helicopter should be able to hover, above the semi-submerged slung load within a distance of 0.50 meters horizontally and 0.50 meters vertically.

Table 4.2: Requirements for the autonomous helicopter with the semi-submerged load attached.

4.3 Acceptance Test Specification

This section will introduce the technical specifications and procedures to validate the requirements presented in section 4.2. As in the requirements, the acceptance tests are partitioned into two main sections: one for each requirement group. The first section will treat the requirements to the helicopter itself. These requirements are presented in table 4.1 on the previous page. After that the requirements to the full system, which are presented in table 4.2, will be addressed.

The acceptance tests presented in this section, have to be completed for each of the control strategies used in this work. The numbering of the acceptance tests should then be numbered as: Acceptance test 1.1, 1.2 and so forth, for each control strategy to be tested.

4.3.1 Specification of Standalone Helicopter Acceptance Tests

HP1 Hovering of the helicopter, in a steady position above a certain point

As stated in the requirements, the helicopter should be able to hover within a distance of 0.50 m horizontally and 0.50 m vertically. This is tested by simulation. The position reference of the helicopter is set to a fixed position and the actual position of the helicopter during the test is compared. The simulation is repeated 10 times with different wind inputs, and each time it is investigated if the position is outside the specified region. If the boundaries of this region are violated in any of the simulations the test is failed, otherwise the test is passed.

HP2 Flying towards a reference position

The helicopter should be able to fly from its current location towards one specified by a reference position. Therefore it should be able to maintain the specified flight velocities with a maximum allowed deviation of 1.00m/s towards the destination. Furthermore, it should be able to transcend from hover to flight in less than 10.00 s, without deviating more than 0.50 m in altitude, defined from when a new reference position is given, until the helicopter velocity is within 1.00m/s of the reference. This is tested by a simulation where the controller is given a reference position, further away than it can reach within the simulation time. The test is repeated 10 times and each time it is investigated if the velocity, transition time and altitude requirements are within the specified limits. If the boundaries are violated in any of the simulations the test is failed, otherwise the test is passed.

HS3 Arriving at a reference position

The helicopter should upon arrival at the reference position be able to change from flight to hover, in less than 10.00 s, while staying within the the altitude deviation limit of 0.50 m. The transition time is defined from when the helicopter is within 5 m of the reference position, until it is within the 0.50 m hover zone. When changed to hover, the helicopter should be able to hover within a distance of 0.50 m horizontally and 0.50 m vertically. This is tested by simulation where the helicopter is given a position reference within the reach of the helicopter during the simulation. The simulation is run 10 times, and must fulfil the requirements at all times to be considered a success.

4.3.2 Specification of Full System Acceptance Tests

SP1 Hovering of the helicopter, in a steady position above the semi-submerged load, with a tense cable

The helicopter should be able to hover, above the semi-submerged slung load within a distance of 0.50 m horizontally and 0.50 m vertically of the specified position. This is tested via a simulation where the helicopter is given a reference position, and the actual position of the helicopter is compared. The vertical position reference of the helicopter is fixed at 3.00 m since that is the length of the slung load cable. This test is run 10 times and each time it is investigated if the position is outside the specified region. If the boundaries of this region are violated in any of the simulations the test is failed, otherwise the test is passed.

SP2 Flying towards a reference position

Apart from hovering with a slung load, the helicopter should also be able to fly from its current location to one specified by a horizontal reference position, while dragging the slung load behind it. The helicopter should therefore be able to transcend from hover to flight in less than 10.00 s without deviating more than 1.00 m in altitude. It should also be able to maintain a velocity towards the reference position with a maximum allowed deviation of 1.00 m/s. This is tested by a simulation where the helicopter is given a reference position outside its reach during the simulation. This is simulated 10 times and each time it is investigated if the requirements are within the given limits. If the limits violated in any of the simulations the test is failed, otherwise the test is passed.

SS3 Preventing fully-submerged load while in flight

The helicopter should at all time during flight generate enough lift for the slung load to keep it from becoming fully submerged. Since the slung load being dragged by the helicopter is an imitation of a person being dragged in the water, that should be kept from drowning. The buoy used in this work, which is simplified to a sphere has a radius of 114.30 mm, which is the vertical position the load should keep above at all times in flight. The results of this test is made based on the data from the flight tests in test **SP2**. If the requirement is not fulfilled in all simulations the test is failed, otherwise it is passed.

SS4 Arriving at a reference position

Upon arrival at the reference position, the helicopter should be able to change from flight to hover, in less than 10.00 s while staying within the limits for vertical position of 0.50 m while doing so. When changed to hover, the helicopter should be able to hover, above the semi-submerged slung load within a distance of 0.50 m horizontally and 0.50 m vertically of the specified position. This is tested by simulation where the helicopter is given a reference position that it can reach within the time span of the simulation. This simulation is run 10 times, and must fulfil the requirements at all times to be considered a success.

Summary of part I

In the problem analysis, some of the significant work that has been done prior to this project has been examined. The field of modelling and control of fully submerged bodies is a well established area. However no work has been found concerning helicopters with semi-submerged slung loads, thus the area of this report seems unexplored.

Following the basic mechanics of small scale helicopters, with focus on the rotor mechanics and functionalities was described.

After this description the case setting of the report was described and narrowed down to the following final problem statement:

How can a UAS that is capable of autonomously towing a semi-submerged slung load through water, be modelled and controlled?

Thereafter specific requirements to the system and following an acceptance test specification have been posted. The acceptance test specification describes the procedures for testing if the individual requirements has been fulfilled.

Part II

Model

Part introduction

Contents

- 5 Helicopter Model
- 6 Semi-Submerged Load Dynamics
- 7 Rigid Body Model
- 8 Model Simulation and Verification
- 9 Model Implementation for Helisim

Purpose

This part is meant to provide a mathematical model to describe the physical behaviour of a slung load, and its interaction with a helicopter. The model for the helicopter itself will not be derived, but rather a complete helicopter model will be adopted.

A block diagram of the model parts that will be derived in this part, and their interactions with each other and the Bisgaard helicopter model, can be seen in picture 4.1. The Bisgaard model is illustrated in grey, to better show that the block is used as it is, and not derived in this work. In the helicopter model introduction, the Bisgaard model block diagram shown in its full extend, and not as one combined block.

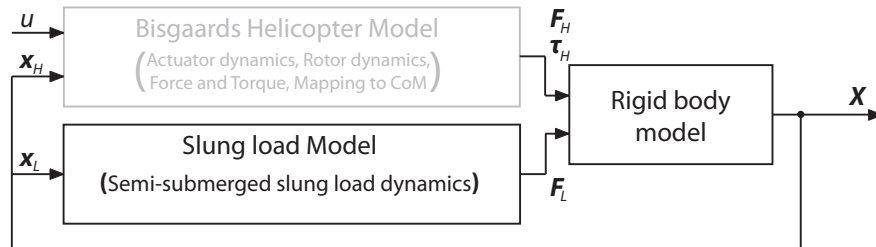


Figure 4.1: A block diagram showing the interactions between the project slung load model and the Bisgaard model.

Goals

- *Examine the Bisgaard helicopter model describing the forces and torques that is to be used in the rigid body modelling.*
- *Derive a model describing the physical behaviour of the slung load.*
- *Derive a model describing the interactions between the two rigid bodies that is the helicopter and the slung load.*
- *Simulate and implement the derived slung load and rigid body models.*

Helicopter Model

The model presented in this chapter is a model concerning the physical behaviour of the AAU-Bergen helicopter, which is presented in chapter 3.1. Originally the model was developed for the purpose of real-time simulation, model based estimation and controller design. This means that it had to be sufficiently complex so that a good controller could be made from it, and it has to be simple enough to be run considerably faster than real-time on the on-board computer in order to be used in the model based estimation.

The fact that the model is designed for the same helicopter, and for the same purpose as it is the aim of this project, makes it highly qualified as a starting point. Figure 5.1 show which parts of the Bisgaard model that will be adopted directly. Mathematical models for the load dynamics and rigid body behaviour will be derived in the following two chapters. As it can be seen in figure 5.1, the purpose of the submerged load model is to calculate the forces exerted on the load, based on the states of the load. The purpose of the rigid body model is to calculate the states of the helicopter and slung load, based on the forces and torques on the helicopter and load. In essence what will be reused from the Bisgaard model is the force dynamics of the helicopter.

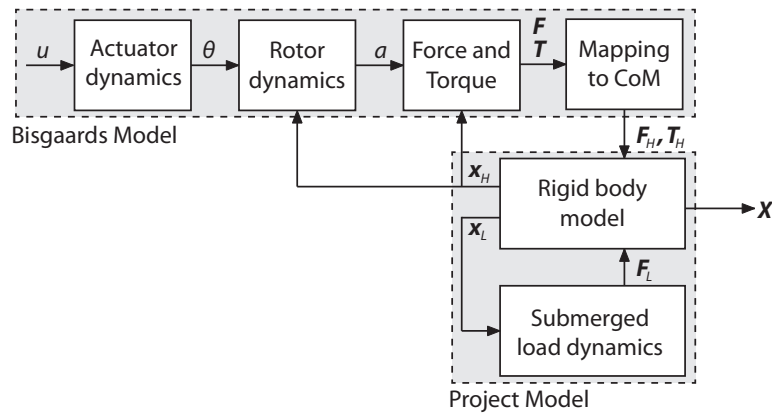


Figure 5.1: A block diagram showing the interactions between the different blocks in the helicopter with submerged slung load model.

5.1 Coordinate Systems

In [7] the model is expressed in five different reference frames, all right handed Cartesian systems. The initial reference frame is fixed with respect to the earth at a position close the the starting point of the helicopter. It is called the earth fixed frame, and is denoted by \mathbb{E} . It is correspondent to the Universal Transverse Mercator (UTM) coordinate system, where the first axis is pointing north, the second is pointing east and the third is pointing down. Two other coordinate systems are fixed to the bodies of the centre of mass (COM) of the helicopter and the slung load respectively. Denoted by \mathbb{H} and \mathbb{L} . The two last frames are positioned in the centre of each of the helicopter rotors, and are aligned to the helicopter fixed frame. The main and tail rotor frames are denoted \mathbb{MR} and \mathbb{TR} respectively. The latter two will not be mentioned further in this work.

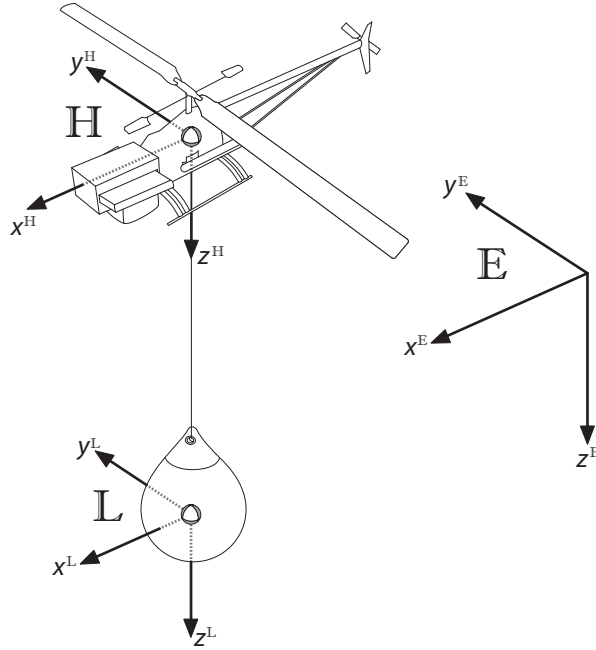


Figure 5.2: An illustration of the three frames used when modelling the helicopter with slung load.

-
- | | |
|--------------|--|
| \mathbb{E} | The earth fixed frame. |
| \mathbb{H} | The helicopter centre of mass fixed frame. |
| \mathbb{L} | The load centre of mass fixed frame. |
-

The earth fixed frame, Helicopter COM frame and Slung load COM frame are all adopted directly, to ease the process of merging the models attained in this work and the helicopter dynamics. An illustration of the three frames can be seen in figure 5.2.

5.2 Forces and Torques

Bisgaard's helicopter model which is presented in [7], introduces the following forces acting on the helicopter:

Force	Description
F_g	<u>Gravitational force</u> : This force is calculated as the mass of the helicopter, multiplied with the gravitational acceleration.
F_{mr}	<u>Main rotor thrust force</u> : This force is a summation of multiple forces, whereof the most dominant one is the lift force. These forces are found by integrating the aerodynamic force along a rotor blade, and then integrating along one rotation, to gain an average force, and finally multiplying with the number of blades. These calculations assume small angles for flapping and blade pitch.
F_{tr}	<u>Tail rotor thrust force</u> : This force is much analogous to the main rotor thrust force, with only the thrust force, so rotor flapping is neglected.
F_h	<u>Helicopter fuselage drag force</u> : Calculated by a standard quadratic drag equation, where the drag is proportional to the incident wind velocity squared. The fuselage force is assumed to generate on torque.
F_{tf}	<u>Tail fin drag force</u> : The tail fin drag force is calculated in the same manner as the fuselage force, where the tail fin is modelled as a flat panel, that only generates drag. A compensation is made to the incident wind velocity, as the fin is partially inside the tail rotor wake.
F_{tp}	<u>Tail plane drag force</u> : The tail plane is modelled in the same manner as the tail fin, but it is not affected by the wake of the tail rotor.
F_{fp}	<u>Front plane drag force</u> : The front plane drag force is calculated in the same manner as the tail fin drag force, but instead of being in the tail rotor wake, it is in the main rotor wake, and a correction is made to the incident wind velocity.

Table 5.1: Forces acting on the helicopter. These are introduced in [7], and will be used later in the rigid body model.

Some of the forces presented in 5.1 give rise to torques. These torques are listed in table 5.2 on the following page along with an additional torque that is not expressed in terms of the presented forces.

Gravity does not contribute to the torque, since it is acting in the COM, and the fuselage force is assumed to not contribute as well.

Torque	Description
τ_{mr}	<u>Main rotor torque:</u> This torque is mainly a result of aerodynamic forces and flapping dynamics of the main rotor. This torque is what makes the helicopter pitch and roll, as a result of the blade pitch. It is calculated by integrating along a blade, as for the main rotor force, then integrating over one revolution and multiplying with the number of blades.
$\mathbf{F}_{mr} \times \mathbf{p}_{mr}$	<u>Torque caused by main rotor thrust:</u> This torque is present since the main rotor thrust force is not located in the centre of mass of the helicopter.
$\mathbf{F}_{tr} \times \mathbf{p}_{tr}$	<u>Torque caused by tail rotor thrust:</u> This torque is present since the tail rotor thrust force is not located in the centre of mass of the helicopter. This torque is the torque responsible for the yaw movement of the helicopter.
$\mathbf{F}_{tf} \times \mathbf{p}_{tf}$	<u>Torque caused by the tail fin drag force:</u> This torque is present since the tail fin drag force is not located in the centre of mass of the helicopter.
$\mathbf{F}_{tp} \times \mathbf{p}_{tp}$	<u>Torque caused by the tail plane drag force:</u> This torque is present since the tail plane drag force is not located in the centre of mass of the helicopter.

Table 5.2: Torques acting on the helicopter, as introduced by [7]. Note that in this report, positions are denoted by \mathbf{p} whereas they are denoted by \mathbf{R} in [7].

5.3 Helisim

The implementation of the Bisgaard model has already been done through a series of Simulink S-functions in a simulation environment from Aalborg University, called Helisim. The simulation is a MATLAB/Simulink implementation of a helicopter model, with the possibility to enable slung load, and a graphical representation of states, on a 3D helicopter, implemented with the Irrlicht graphics engine. A illustration of the Simulink representation of Helisim is shown in figure 5.3. Helisim includes, apart from the helicopter and a slung load model, sensor dynamics, sensor fusion on the on-board computer, and the possibility to use both LQ and PID controllers to control the helicopter as specified in an optional mission file.

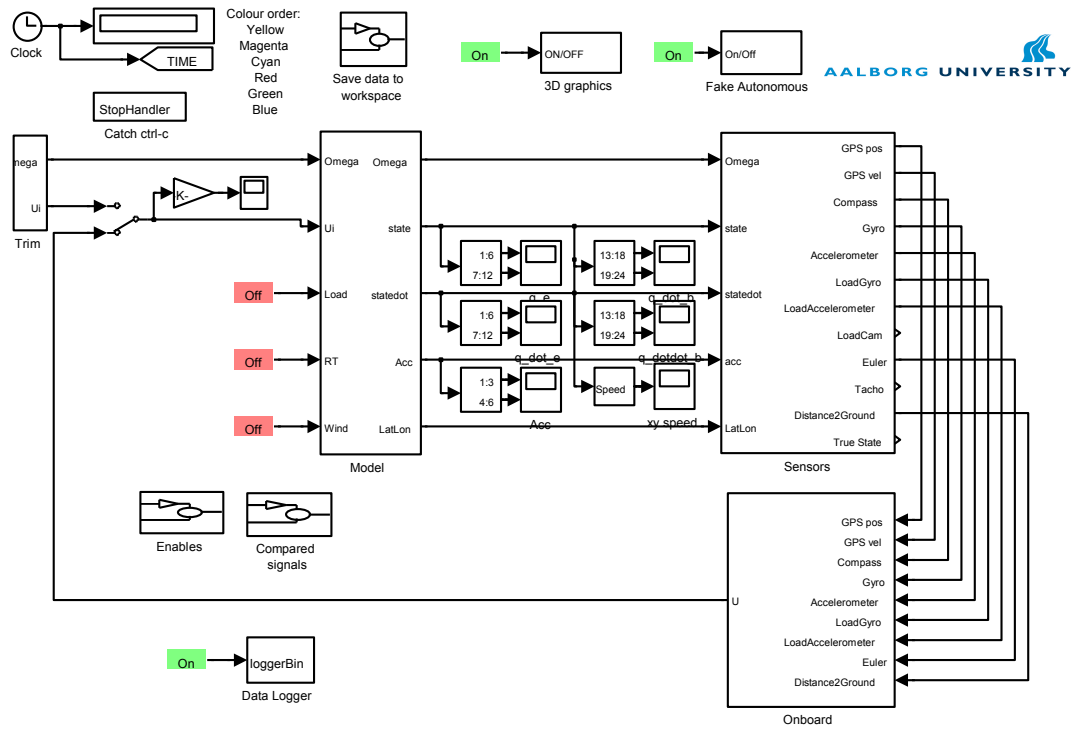


Figure 5.3: An illustration of the Helisim implementation of the Bisgaard model, also including sensor dynamics, sensor fusion and LQ and PID controllers.

Helisim will be used in the remaining parts of this work, for simulations where a helicopter model is needed, as well as in the control part III where the computed controllers will be implemented and tested in Helisim. The estimated sensor values from Helisim is used in the acceptance test in chapter 13, to test the performance of the final controllers.

During this work the source files for Helisim will be modified, to fit the studied case. As described earlier in this chapter, the force generation of the helicopter it self is used as it is, but new functions are written for the rigid body model of the combined system, and force generation of the load.

The implementations of the functions can be found in the two files: `BuoyRigidBodyModel.cpp` and `BuoyLoadFunctions.cpp`, which can be found on the enclosed DVD:



`/Source/BuoyRigidBodyModel.cpp`

`/Source/BuoyLoadFunctions.cpp`

5.4 Helicopter model conclusion

With the general setup of the helicopter model in the beginning of the chapter, it has been made clear what parts of the helicopter with load system that can be adopted directly from the Bisgaard model, and which parts should be produced in this work. The general structure also show how the modelling part of this work should be interfaced with the existing helicopter model, where the relevant forces and torques needed for the rigid body modelling, connecting the helicopter model and the slung load model, is presented earlier in the chapter.

The inertial earth fixed frame and the noninertial helicopter and slung load frames of the bisgaard model is also adopted and used in the remaining part of the modelling part, to make implementation easier.

Semi-Submerged Load Dynamics

This chapter will focus on the derivation of the dynamic slung load model which, as shown in figure 6.1, is the mapping from the states of the system to the forces and torques applied on the helicopter.

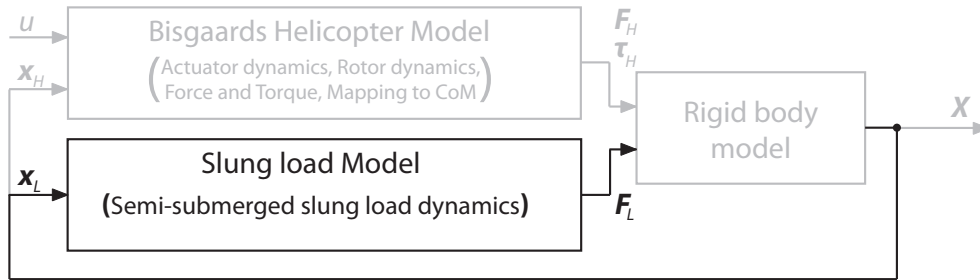


Figure 6.1: A block diagram showing the interactions between the submerged slung load dynamics block and the other parts of the model.

The content of this chapter is inspired by [2]. Which is a book dealing with autonomous underwater vehicles and modelling of these. While the book does not directly cover semi-submerged load modelling, it does however include the hydrodynamics needed for the load modelling.

First the static effects i.e. gravity and buoyancy will be treated, then an expression for the added mass will be derived, and lastly expressions for the lift and drag forces will be found. These effects will constitute the semi-submerged part of the slung load model. The verification of the model is done in the simulation and verification chapter 8 on page 55.

6.1 Hydrostatic Effects

This section will introduce the hydrostatic effects that act on the load. The load is, like all other things on earth, subjected to gravity, which is described by newtons 2nd law of motion as shown in (6.1). Provided that the load is always subjected to gravity, there must be an equally opposing force counteracting gravity, if the load is static. This force is called buoyancy is described by Archimedes' principle [15] which is shown in (6.2).

$$\mathbf{F}_g = m_l \mathbf{g} \quad (6.1)$$

\mathbf{F}_g	Gravitational force.	N
m_l	Mass of the load.	kg
\mathbf{g}	Gravitational acceleration .	m/s ²

$$\mathbf{F}_b = \rho V_d \mathbf{g} \quad (6.2)$$

\mathbf{F}_b	Buoyancy force acting on the load.	N
ρ	Mass density of displaced water.	kg/m ³
V_d	Volume of the displaced water.	m ³

As the load is considered a sphere, the buoyancy is not a linear function of depth. Figure 6.2 represent the value of the buoyancy and gravity magnitude over the depth.

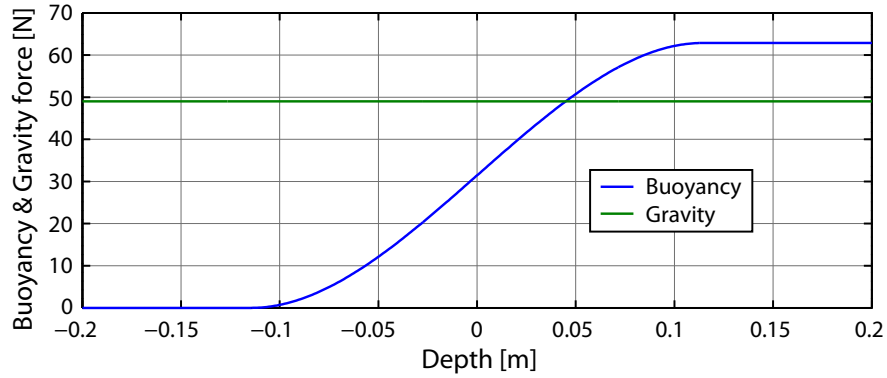


Figure 6.2: Representation of the buoyancy and gravity magnitude over depth. The mass used for this plot is 5 kg, the load radius is $114.3 \cdot 10^{-3}$ m, the water density is 1025 kg/m^3 and the gravitational acceleration is 9.82 m/s^2 .

It can be seen in figure 6.2, there is an equilibrium point. This equilibrium point located at a depth of 0.045 m is given by the relationship $\rho V_d = m_l$. This equilibrium point is 0.045 m below the water surface, for the given parameters.

6.2 Added Mass

The motion of a body in a fluid induces acceleration of the fluid surrounding it. Accelerating the fluid requires a force, equivalent to the mass of the accelerated fluid multiplied with the acceleration induced to the fluid. For bodies in air, this effect can be neglected as the density of air is, most of the time, very small compared to the density of the considered body. However for a body moving in water, the density of the body usually is very similar to density of the fluid it is moving in, therefore this effect must be taken into account.

The hydrodynamic force can be seen as a function of the velocity and acceleration of the body, as explained by (6.3).

$$\mathbf{F}_{\text{hydro}} = \mathbf{F}_{\text{damp}}(\mathbf{u}) + \mathbf{F}_{\text{mass}}(\dot{\mathbf{u}}) \quad (6.3)$$

\mathbf{F}	Hydrodynamic forces.	N
\mathbf{u}	Relative velocity of the load with respect to the fluid.	m/s

The velocity dependent part of the expression is explained in section 6.3 on page 33, and the remaining part of this section will focus on deriving the part, which is dependent on the acceleration.

The hydrodynamic force, dependent on the mass of the accelerated fluid is often added into the dynamic equations, by means of an increase in the mass of the body itself, and this is convenient, as the force is usually proportional to the acceleration of the body. Hence the name added mass, or virtual mass. The exact value of this force would be the acceleration times the mass of each fluid particle affected by the motion of the body [2]. This is not realistic to compute, hence an approximation is made by integrating the hydrodynamic pressure over the entire surface of the body, and isolating the acceleration dependent part. The pressure is described by Bernoulli's equation for irrotational unsteady potential flow, which is shown in (6.4).

$$P = -\rho \left(\dot{\phi} + \frac{1}{2} |\nabla \phi|^2 \right) \quad (6.4)$$

P	Hydrodynamic pressure on a submerged point.	Pa
ρ	Density of the fluid.	kg/m ³
ϕ	Velocity potential, such that $\mathbf{u} = \nabla \phi$.	m ² /s

This equations is used in the field of modelling ocean surface waves, and impact on structures floating on it [11]. According to [21] the velocity potential around a sphere can be described as:

$$\phi = |\mathbf{u}| \cos(\theta) \frac{r}{2} \quad (6.5)$$

θ	Half of the apex angle, spanned by the spherical section. I.e. the angle between \mathbf{u} and the line through the centre of the sphere and a point on the surface.	rad
r	Radius of the load.	m

Using this and making the substitution $\mathbf{u} = \nabla\phi$ in (6.4) results in the following expression of the pressure, as shown in (6.6), and illustrated in figure 6.3.

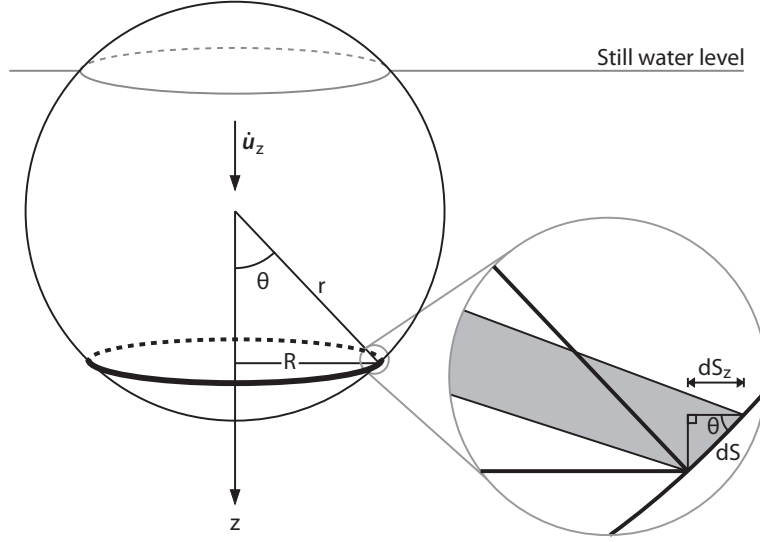


Figure 6.3: Illustration of the integration of the pressure on the sphere.

$$P(\theta) = -\frac{\rho}{2} \left(|\dot{\mathbf{u}}| r \cos(\theta) + |\mathbf{u}|^2 \right) \quad (6.6)$$

Hence the expression of the hydrodynamic force is obtained, by integrating the pressure over the submerged surface.

$$F_{\text{hydro}} = \int_0^{\cos^{-1}(-h/r)} P(\theta) \cdot 2\pi r^2 \cdot \sin(\theta) \cos(\theta) d\theta \quad (6.7)$$

h	Vertical displacement of the centre of the sphere relative to the water level, positive downwards.	m
-----	--	---

The upper limit is provided by the submersion depth of the body inside the fluid, and by substituting expression (6.6) in (6.7) and performing the integration, (6.8) is obtained:

$$\mathbf{F}_{\text{hydro}} = -\rho\pi r^2 \left(\frac{1}{3} |\dot{\mathbf{u}}| r \left(\frac{h^3}{r^3} + 1 \right) - \frac{1}{2} |\mathbf{u}|^2 \left(\frac{h^2}{r^2} - 1 \right) \right) \quad (6.8)$$

Comparing (6.8) to (6.3), the expression of the added mass-force is obtained in (6.9).

$$\mathbf{F}_{\text{mass}} = -\frac{\rho\pi}{3} (h^3 + r^3) |\dot{\mathbf{u}}| \quad (6.9)$$

The coefficient in front of the acceleration ($\ddot{\mathbf{u}}$) is called added mass induced by the displacement of water in contact with the sphere. As a result, this effect can be seen as a dynamic modification of the mass of the body.

Figure 6.4 shows the value of the added mass over depth. The added mass is not a linear function of the depth, due to the spherical shape of the buoy. Moreover, a correction is needed as the results are only valid for a submersion depth of \pm the radius of the sphere, i.e. from not submerged to fully submerged. Outside this region the added mass is kept constant.

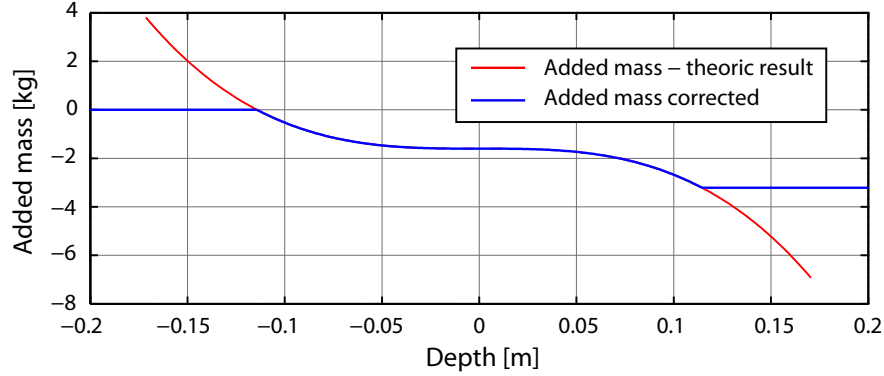


Figure 6.4: Representation of the added mass over submersion depth, showing the mathematical result and the boundary corrected result.

6.3 Damping Effects

Damping effects are mainly due to viscosity of the fluid. According to [2], a good simplification is to keep only linear and quadratic dissipation effects. Due to the shape of the load and to the absence of waves, coupled damping terms are zeros.

It is possible to model the action of the viscous effect with two forces: the drag force that is parallel to the velocity of the body in the fluid, and the lift force that is perpendicular to the drag force. These forces are explained in (6.10) and (6.11)[2].

$$F_{\text{drag}} = \frac{\rho_w U^2 S C_d(Re)}{2} \quad (6.10)$$

F_{drag}	Drag force.		N
ρ_w	Density of water.	1025	kg/m ³
U	Velocity of the body.		m/s
S	Projected surface area.		m ²
C_d	Drag coefficient.		
Re	The Reynolds Number: Defined in (6.12).		

$$F_{\text{lift}} = \frac{\rho_w U^2 S C_l(Re, \alpha)}{2} \quad (6.11)$$

F_{lift}	Lift force.	N
C_l	Lift coefficient.	
α	Angle of attack.	rad

In (6.10) and (6.11) the Reynolds number, which is defined as (6.12), is not directly used in the calculations of the lift and drag forces, but rather being a description of how the flow of the fluid around the buoy is expected to be.

$$Re = \frac{\rho_w |U| L}{\mu} \quad (6.12)$$

μ	Dynamic viscosity of the fluid.	Pa s
L	Characteristic length.	m

The lift and drag equations presented in (6.10) and (6.11) are not sufficient for this work, since they describe the lift and drag forces applied on a fully submerged load. In this work, where a semi-submerged slung load is used, only the part of the load which is submerged has to be taken into account. This dependency is examined further in section 6.4.

The motion of a body in a fluid disrupts its flow, and introduce vortices in the fluid behind the body. These vortices come from the shape of the body, and is affected by the rotation of the body. Considering that the shape of the actual body is close to a sphere no vortexes can arise from rotations of the body. Therefore, these disturbances will be considered negligible as they are very small with respect to lift and drag forces.

6.4 Lift and Drag Force Corrections and Simulations

The purpose of this section is to get the values for the lift and drag forces at different velocities and depths through simulation, and include possible depth and angle of attack dependencies in the lift and drag coefficients in the standard lift and drag force formulas.

To get the lift and drag forces on the buoy with different velocities and depths, a simulation tool called *SolidWorks 2010* can be used. SolidWorks offers a *Computational fluid dynamics* (CFD) tool, that simulates the flow of fluids inside or around an object.

6.4.1 SolidWorks Flow Simulations and Results

Figure 6.5 on the next page shows an illustration of the set-up in SolidWorks. The sphere is the buoy with its size, material and density entered. The large box is the computational domain in which the fluid is placed. The fluid, being water, has in this series of simulations been set to flow in the direction of the arrow, with a velocity

set for each of the simulations. The fluid trajectory and surface pressure are parts of the results from a flow simulation, along with the set goals, here being the x- and z-component of force, translated to the drag and lift forces. All the simulations were done without gravity enabled, which means that the buoyancy is also disabled, due to buoyancy being a direct effect of gravity. The example simulation shown in figure 6.5 were computed with the computational domain covering half the buoy, translated to a top boundary of the box at 0 cm, and a water velocity of -4 m/s with the value being negative since the wanted forces are those of the buoy moving through the water, simulated as the water moving around a stationary sphere.

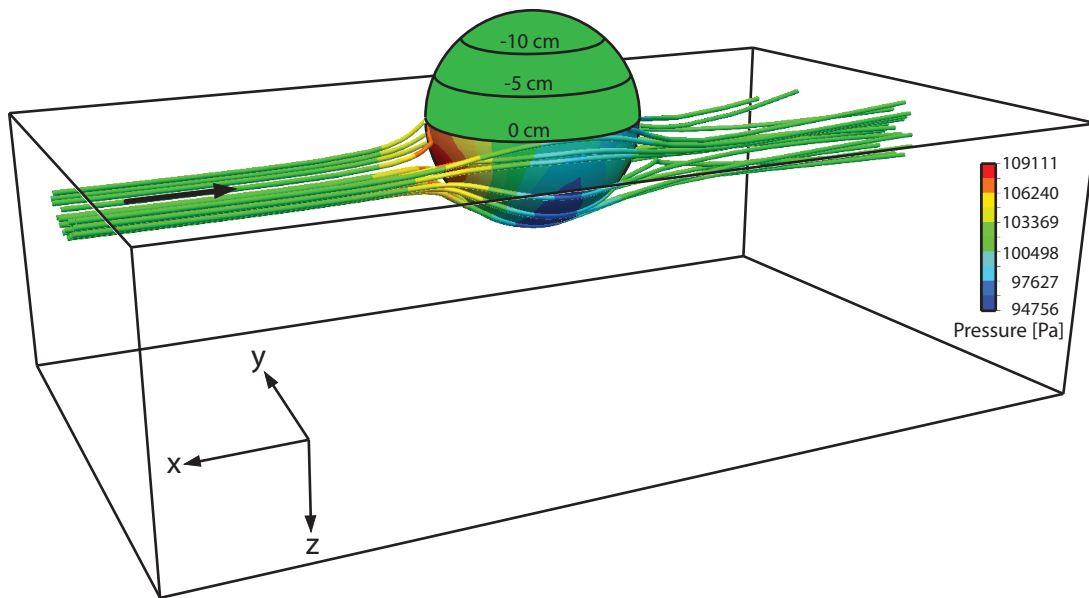


Figure 6.5: An illustration of the Flow simulation tool in SolidWorks 2010 with the fluid box, surface pressure plot, and fluid trajectory.

A series of simulations is run to find the lift and drag forces, at different depths and velocities. The ranges and step sizes are chosen to be as listed below. The height of the buoy is defined at 0 cm being half of the buoy submerged and 0,114 m, which is the radius of the sphere, is then completely submerged. The additional point inserted with the buoy completely submerged, is to examine the difference in lift and drag forces when the buoy stop having contact with the surface.

Water velocity	1 m/s steps. Range 0 ; 5 m/s.	
Water level	5 cm steps. Range -0.1 ; 0.15 m	Additional point: 0.114 m

The simulations data and a MATLAB-script for plot generation, can be found on the DVD:



/Model/Hydro dynamics/lift_drag_3d_plots.m
/Simulation/Solidworks/depth/

Figures 6.6 and 6.7 shows the drag and lift forces as a function of velocity and buoy depth. The lift values are positive in figure 6.7 due to the \mathbb{E} frame being defined with a z-axis pointing downwards to be equal to the rest of the defined frames.

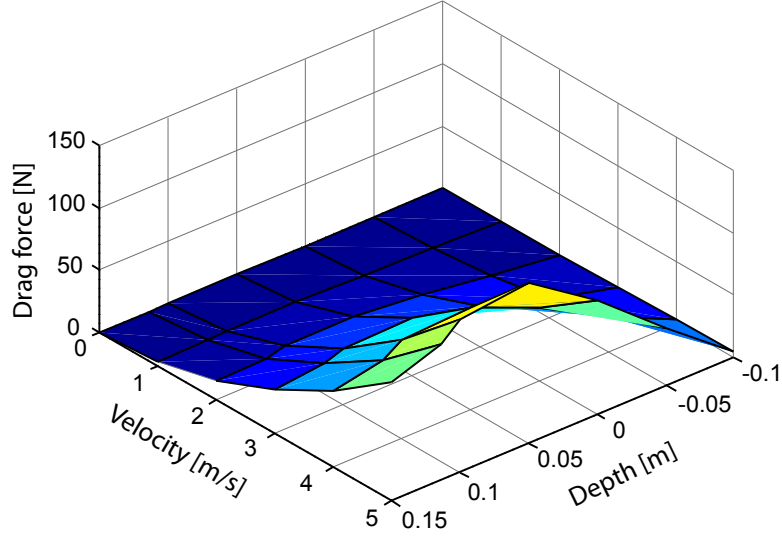


Figure 6.6: 3D plot of the drag force on the sphere at different velocity and with different amounts of the load submerged.

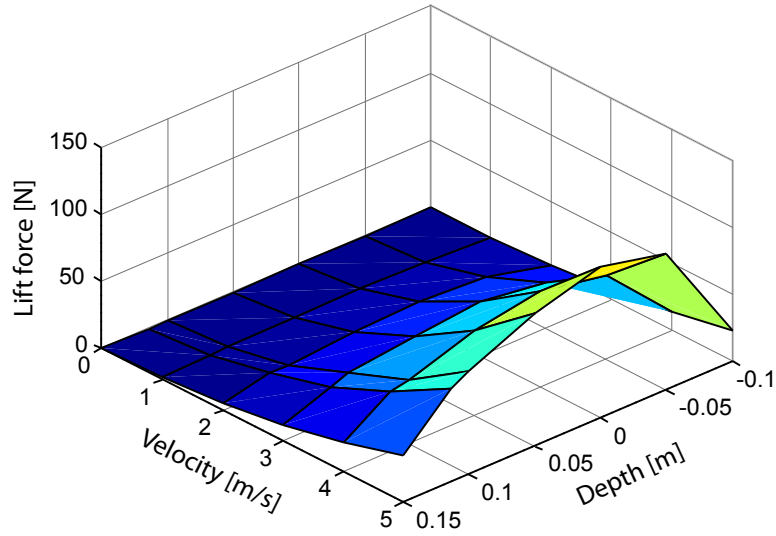


Figure 6.7: 3D plot of the lift force on the sphere at different velocity and with different amounts of the load submerged.

The two 3D plots show how the lift and drag forces have some non-linearities from the velocity and depth dependencies that have to be accounted for, with the velocity dependency already included in the standard lift and drag force equations.

6.4.2 Lift and Drag Force Calculation with Depth Dependency

The lift and drag forces have dependency of the buoy depth in water, based on the illustrations in figures 6.6 and 6.7. This dependency is however not included in the formulas for lift and drag introduced in (6.10) and (6.11) on page 34.

The depth dependency of the lift and drag forces emerge from the lift and drag coefficients changing, depending on the amount of the buoy submerged. To find the relation, the lift and drag coefficients are calculated for all of the simulations, except those with a velocity of 0 m/s, since that would only result in coefficients of 0. As seen from (6.13), both formulas have been rewritten from (6.10) and (6.11) to isolate C_D and C_L . Using the simulated data, the only unknown parameter is the surface area, which is calculated using the methods shown in appendix A on page 175.

$$C_D = \frac{2F_{drag}}{\rho_w U^2 S} \quad C_L = \frac{2F_{lift}}{\rho_w U^2 S} \quad (6.13)$$

Table 6.1 and 6.2 show the calculated values of C_D and C_L for all the simulations with a velocity above 0 m/s. From the tables it is seen that the C_D and C_L values, differ over the depth columns.

$C_D \cdot 10^{-3}$		Velocity [m/s]				
		1	2	3	4	5
Depth [mm]	150.0	190.5	186.7	185.0	183.8	183.1
	114.0	203.5	200.0	209.9	208.0	207.8
	100.0	240.8	248.3	243.4	242.9	240.7
	50.0	320.2	321.6	319.2	318.6	318.5
	0.0	355.2	356.8	352.6	351.1	352.1
	-50.0	367.7	369.0	362.1	366.7	382.4
	-100.0	333.5	324.2	331.9	328.8	337.3

Table 6.1: Drag coefficient table, calculated for each of the simulations.

$C_L \cdot 10^{-3}$		Velocity [m/s]				
		1	2	3	4	5
Depth [mm]	150.0	67.9	66.8	64.0	66.3	65.9
	114.0	155.5	150.7	134.5	133.3	133.0
	100.0	157.4	147.0	164.5	144.1	149.1
	50.0	244.5	250.6	244.2	244.2	245.3
	0.0	423.0	437.4	426.5	426.4	429.7
	-50.0	762.6	788.4	775.0	791.8	830.5
	-100.0	1456.3	1491.4	1555.9	1557.9	1615.4

Table 6.2: Lift coefficient table, calculated for each of the simulations.

A mean of the coefficients for all the velocities at each depth is taken, resulting in the vectors presented in (6.14). Depths going from -0.10 to 0.15 m.

$$C_{D,\text{mean}} = \begin{bmatrix} 331.1 \\ 369.6 \\ 353.6 \\ 319.6 \\ 243.2 \\ 205.9 \\ 185.8 \end{bmatrix} \cdot 10^{-3} \quad C_{L,\text{mean}} = \begin{bmatrix} 1535.4 \\ 789.7 \\ 428.6 \\ 245.7 \\ 152.4 \\ 141.4 \\ 066.2 \end{bmatrix} \cdot 10^{-3} \quad (6.14)$$

The coefficient means for each depth form a set of points that can be fitted to a function, using MATLAB and `cftool` from the *Curve Fitting Toolbox*. The coefficient mean points and the fitted functions are shown in figures 6.8 and 6.9. The 15 cm mean points are not used in the fitting of the drag coefficient, to get a better fit for the points where the sphere is not yet completely submerged. The function chosen for the drag coefficient is a second order polynomial, and for the lift coefficient it is an exponential.

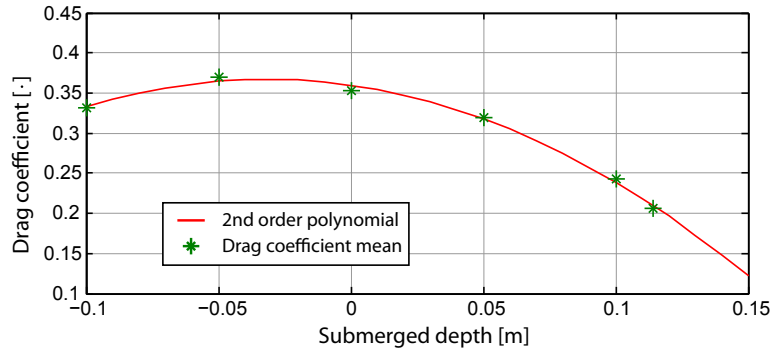


Figure 6.8: A graph showing the second order polynomial fitted to match the drag coefficient mean points.

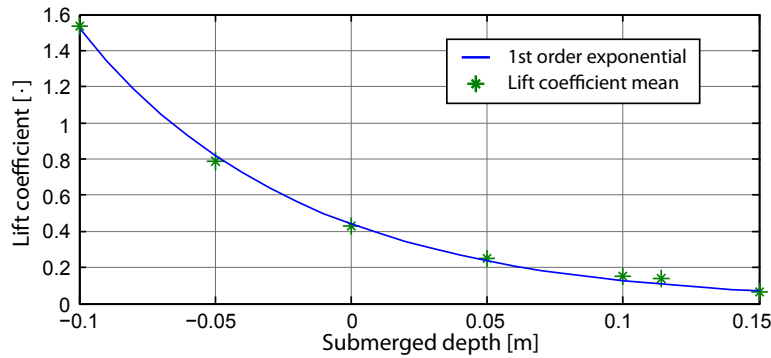


Figure 6.9: A graph showing the exponential fitted to match the lift coefficient mean points.

The fitted functions from `cftool` is shown in (6.15) and (6.16).

$$f_{\text{drag}}(h) = -7.4304h^2 + 0.4693h + 0.3598 \quad (6.15)$$

$$f_{\text{lift}}(h) = 0.4408 e^{-12.39h} \quad (6.16)$$

f_{drag}	Polynomial fitted to drag coefficient means.	
f_{lift}	Exponential fitted to lift coefficient means.	
h	Depth defined from centre of the sphere.	m

Having the two functions fitted to the lift and drag coefficients depending on depth, these can now replace C_D and C_L in the lift and drag force formulas as shown in (6.17) and (6.18).

$$F_{\text{drag,depth}} = \frac{1}{2}\rho U^2 S f_{\text{drag}}(h) \quad (6.17)$$

$$F_{\text{lift,depth}} = \frac{1}{2}\rho U^2 S f_{\text{lift}}(h) \quad (6.18)$$

$F_{\text{drag,depth}}$	The equation of the drag force with a depth dependency.	N
$F_{\text{lift,depth}}$	The equation of the lift force with a depth dependency.	N

With numbers from (6.15) and (6.16) inserted into (6.17) and (6.18), a set of equations for lift and drag forces with a height dependency fitted to the exact buoy used in this work has hereby been derived to be used in the further parts of this work.

The calculations of the lift and drag coefficients using the simulated forces, including the derivation and function of the new lift and drag equations with depth equations, can be found on the DVD:



/Model/Hydro dynamics/lift_drag_differences.m
/Model/Hydro dynamics/lift_drag.m

6.4.3 Lift and Drag Force Calculation with Angle of Attack Dependency

While the difference in the drag and lift forces, as a function of depth have been taken into account in the previous subsection, the changes in lift and drag forces as a function of the angle of attack between the water surface and the relative velocity of the buoy, still needs to be examined. This relative velocity is introduced because the buoy can be moving up and down while being dragged, changing the direction of the actual velocity seen by the buoy. The relative velocity will then be composed of the velocity the buoy is being dragged with, and the up- or downwards velocity enforced by buoyancy and

gravity.

This angle of attack is denoted α . Figure 6.10 shows the set-up in SolidWorks while having an angle of attack that is not equal to 0 rad, in the example case the α angle is $\pi/4$ rad.

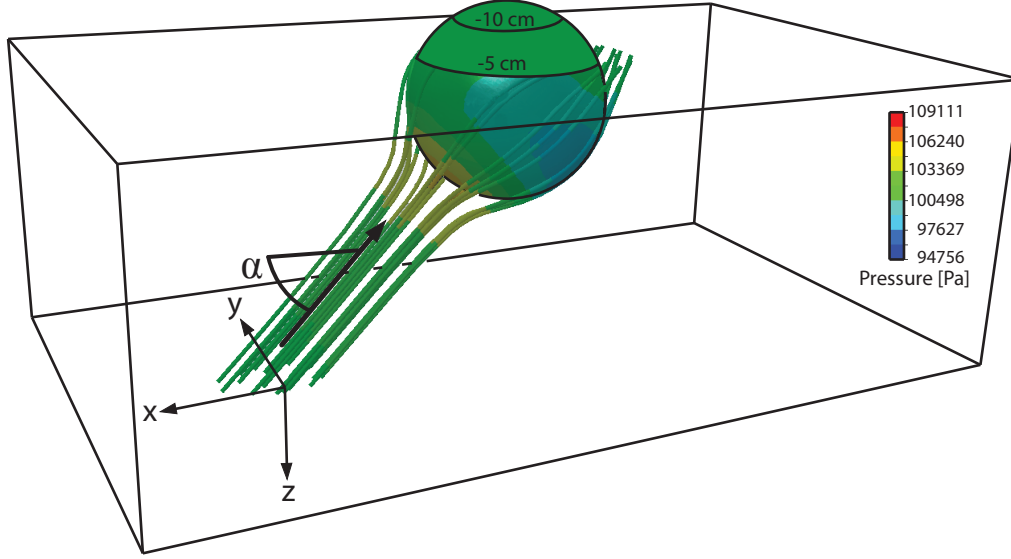


Figure 6.10: An illustration of the flow simulation in SolidWorks with an angle of attack α of $\pi/4$ rad.

While not being dragged by the helicopter, the buoy is expected not to have any movement in the xy -plane of the earth fixed frame. It can however still have movements going up and down in along the z -axis, in terms of the angle of attack denoted α , these movements can be translated to α angles of $\pi/2$ rad.

Lift and drag forces are, like with the depth dependency, calculated using SolidWorks 2010, this time however with the velocity set along the z -axis instead the x -axis used in the depth simulations. The change in lift and drag forces over depth should however still be taken into account here in the angle dependency subsection, since the results from the previous subsection might not be consistent with the results from the simulations with α set to 0 rad. Figure 6.11 show an illustration of how α is defined, as well as the direction of the lift and drag forces.

Since it is shown in tables 6.1 and 6.2 that the lift and drag coefficients vary more over changes in depth, compared to changes in velocity, the simulations done for α dependency is only done at a fixed velocity, with the depth changing between simulations. Table 6.3 shows the drag coefficient calculated at an angle of attack set to $\pi/2$ rad. The lift coefficient is defined to be 0 while the angle of attack is $\pi/2$ rad.

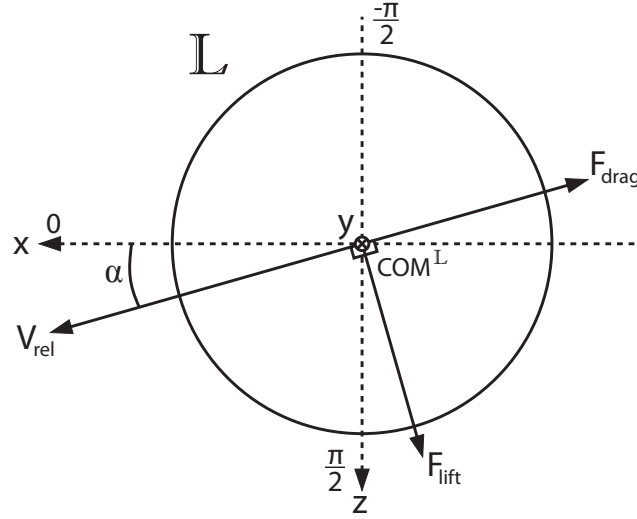


Figure 6.11: Illustration of α and directions of lift and drag forces.

$\alpha = \frac{\pi}{2}$	Depth [mm]						
	-100.0	-50.0	0.0	50.0	100.0	114.0	150.0
$C_D \cdot 10^{-3}$	583.5	344.4	203.3	167.2	218.4	237.8	241.3
$C_L \cdot 10^{-3}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 6.3: Drag and lift coefficient table, calculated for each of the simulations at $\alpha = \pi/2$ rad.

As with $\alpha = 0$ rad, a second order polynomial can be fitted to the points using MATLAB, resulting in the polynomial shown in (6.19).

$$f_{\text{drag},\pi/2}(h) = 19.0162h^2 - 1.8436h + 0.2058 \quad (6.19)$$

Figure 6.12 shows the new fitted polynomial and the points it is fitted to, as well as the original depth dependent drag coefficient. The figure shows that the depth dependent drag coefficient changes drastically with the angle of attack as well.

The new fitted polynomial can then be inserted into the formula for drag force. Resulting in (6.20).

$$F_{\text{drag},\alpha} = \frac{1}{2}\rho U^2 S f_{\text{drag},\pi/2} \quad (6.20)$$

To work the α -angles between 0 rad and $\pi/2$ rad into the equations, a scaling between the two fitted polynomials is used, multiplying $\cos(\alpha)$ on F_{drag} and $\sin(\alpha)$ on $F_{\text{drag},\alpha}$. Regarding the lift force, that goes towards 0, $\cos(\alpha)$ is multiplied onto $F_{\text{lift,depth}}$, as in (6.21) and (6.22).

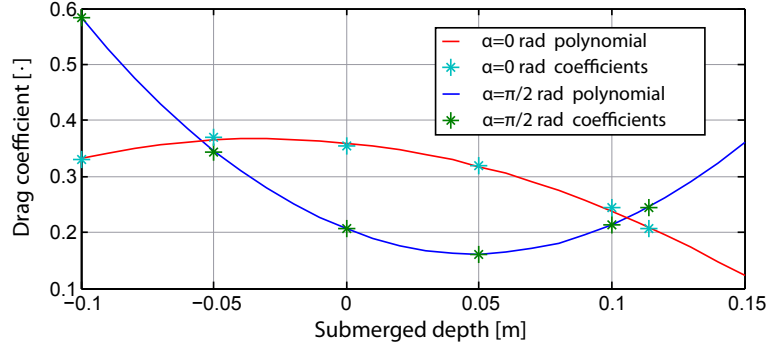


Figure 6.12: A graph showing the second order polynomials fitted to match the drag coefficients at $\alpha = 0$ rad and $\pi/2$ rad.

$$F_{drag}(h, \alpha) = \left(\frac{1}{2}\rho U^2 S f_{drag}\right) \cdot \cos(\alpha) + \left(\frac{1}{2}\rho U^2 S f_{drag, \pi/2}\right) \cdot \sin(\alpha) \quad (6.21)$$

$$F_{lift}(h, \alpha) = \left(\frac{1}{2}\rho U^2 S f_{lift}\right) \cdot \cos(\alpha) \quad (6.22)$$

Figure 6.13 shows a graph of the final lift and drag force equations as a function of the angle of attack. The example plot shown in the figure is done with a depth of 0 cm, velocity of 3m/s and a projected area changing with the angle as shown in appendix A.

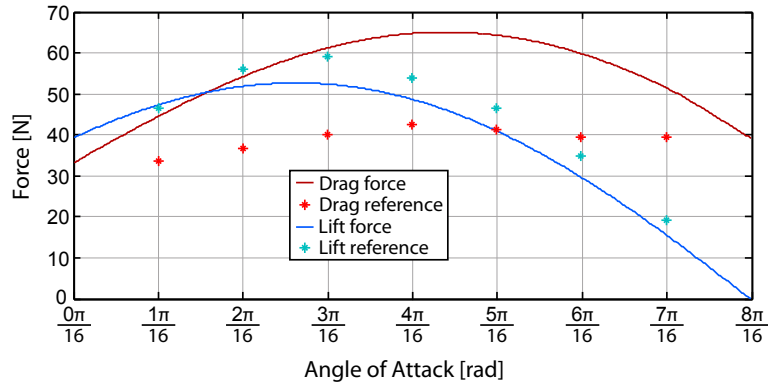


Figure 6.13: A graph showing the calculated lift and drag forces over an angle of attack range from 0 rad to $\pi/2$ rad, along with reference simulation values for every $\frac{\pi}{16}$.

Apart from an example from the final formula for lift and drag forces, figure 6.13 also includes a series of simulated values for the angle of attack changing in $\frac{\pi}{16}$ rad steps, with the reference simulation values shown in table 6.4.

	Angle [rad]						
	$\frac{1\pi}{16}$	$\frac{2\pi}{16}$	$\frac{3\pi}{16}$	$\frac{4\pi}{16}$	$\frac{5\pi}{16}$	$\frac{6\pi}{16}$	$\frac{7\pi}{16}$
F_D	33.571	36.557	40.037	42.581	40.529	39.580	39.422
F_L	46.632	56.042	58.984	53.969	46.506	34.963	19.192

Table 6.4: Drag and lift force table, calculated for each of the reference simulations between $\alpha = 0$ rad through $\pi/2$ rad.

Figure 6.13 shows that the calculated lift force from the example with a particular velocity and depth, follow the reference simulations relatively well, with a lower calculated value throughout the range. The drag force however, does in this particular example not follow the reference simulations very well. As can be seen from table 6.4, the drag force does not vary much throughout the α angle range in the reference simulation values, but it does in the calculated values. This error can come from the simplification of going from the fitted polynomial at an angle of attack at 0 rad to the fitted polynomial at $\pi/2$ rad, using a transition from one to the other with cosine and sine multiplied on their contributions to the calculated lift and drag forces. The derived drag formula will be used in the model verification and final model, even though the calculated value for drag force does not fit entirely with the reference simulation values, since the angle of attack have well fitted values at each end of the range.

Figure 6.14 and 6.15 show the final drag and lift forces, and how they vary over depth and angle. The plots are made for a velocity of 1m/s, which means that the values will scale by velocity squared compared to the ones shown in the plots.

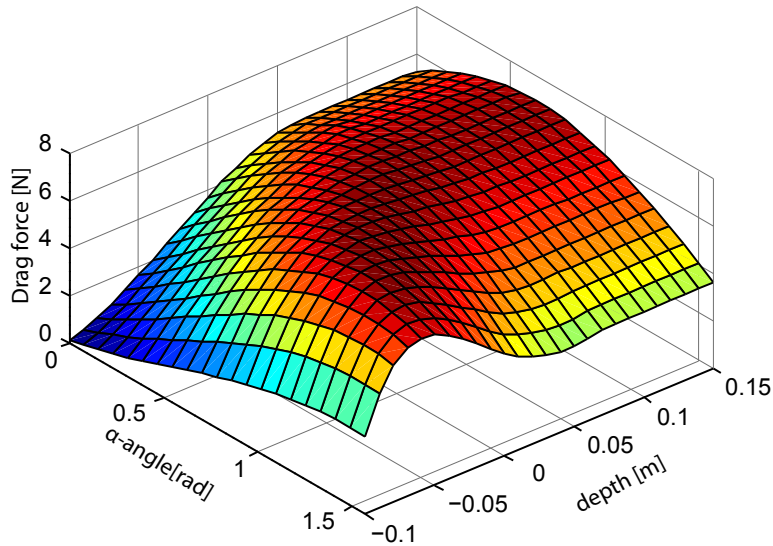


Figure 6.14: 3D plot of the drag force on the sphere at different depths and angles.

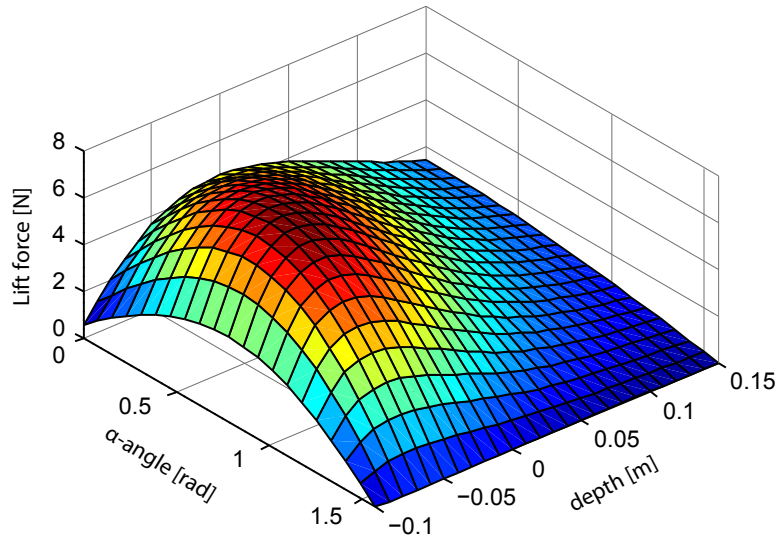


Figure 6.15: 3D plot of the drag force on the sphere at different depths and angles.

The final lift and drag force equations are used in section 8.1. The final function can be found on the DVD along with the simulations used to calculate the angle of attack dependency.



/Model/Hydro dynamics/lift_drag.m
/Simulation/Solidworks/angle/

6.5 Semi-submerged Load Conclusion

The hydrostatic effect section covered the gravity and buoyancy that acts on the load when it is floating in water, while the added mass section included one of the more important hydrodynamic effects, which is the mass of the fluid being dragged with the load, when it is moving. The added mass section also include a depth dependency, since the added mass only should be calculated for the submerged part of the load.

The damping effects introduce the lift and drag forces acting on the load when it is moving, and it is further examined in the lift and drag force correction and simulations that both forces should include both a depth dependency, since the effects from the air is negligible when compared to the effects from the water. Since the slung load have both horizontal and vertical movement, it is concluded that the lift and drag forces should include an angle dependency as well, where a scaling between simulation results for vertical and horizontal movement is presented.

Rigid Body Model

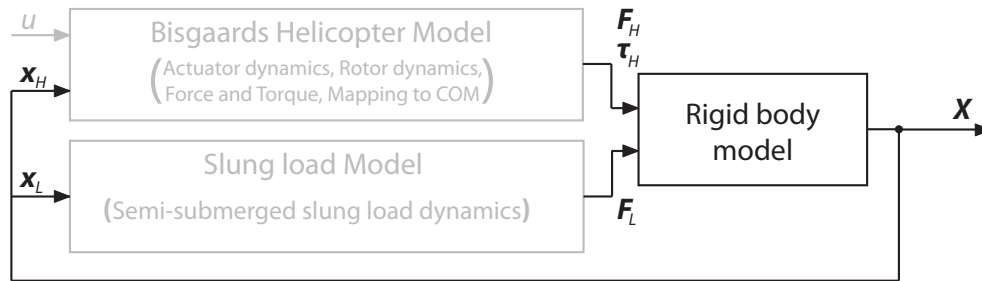


Figure 7.1: A block diagram showing the interactions between the rigid body model block and the other parts of the model.

This chapter will introduce a mathematical model of the dynamics of the helicopter and slung load as well as their mutual influence. This model is intended to translate forces and torques acting on the helicopter and slung load into linear and angular positions and velocities. The model will be derived under three main modelling simplifications. The first of which is that the slung load is considered a point mass. Doing so is reasonable provided that the heading of the load is unnecessary to calculate the forces on the load which is derived in chapter 6 on page 29, and because of the geometry of it this is indeed the case. The second simplification is that the load is connected to the helicopter by a straight, rigid, massless rod. This implies that the load is always a fixed distance to the helicopter. During flight this is not a problem, since the cable will be tense because of the relatively high drag on the load. The third simplification is that the attachment point on the helicopter is positioned in the helicopters centre of mass. This simplification is done since the distance to the attachment point is small, compared to the distance to the load. This implies that the load will have no effect on the heading of the helicopter.

In the following sections the variables used to describe the state of the system is introduced, and the mathematical model based on Newtonian physics will then be derived.

7.1 Degrees of Freedom and State Variables

The helicopter by itself can be seen as a 6 Degrees Of Freedom (DOF) body, three for position and three for orientation, as described in chapter 2 on page 5. The slung load can be perceived in the same manner resulting in 12 DOFs. Considering the load as a point mass and interconnecting the two bodies by a cable reduces the number of DOFs to 8, since if the position of the load has a fixed distance to the helicopter and can be described by two consecutive rotations. These rotations are defined as a rotation about the X-axis followed by one about the Y-axis in \mathbb{E} . These angles are denoted θ_l and ϕ_l respectively. Table 7.1 lists the grouping, naming and a short description of each of the 8 DOFs.

${}^{\mathbb{E}}\mathbf{p}_h = \begin{bmatrix} x_h \\ y_h \\ z_h \end{bmatrix}$	${}^{\mathbb{E}}\mathbf{p}_h$ describes the position of the helicopter as seen in the earth-fixed frame by the Cartesian coordinates $\{x_h \ y_h \ z_h\}$.
${}^{\mathbb{E}}\boldsymbol{\theta}_h = \begin{bmatrix} \theta_h \\ \phi_h \\ \psi_h \end{bmatrix}$	${}^{\mathbb{E}}\boldsymbol{\theta}_h$ describes the orientation of the helicopter as seen in the earth-fixed frame by the Euler angles $\{\theta_h \ \phi_h \ \psi_h\}$. Note that $\boldsymbol{\theta}_h$ is not equivalent to θ_h .
${}^{\mathbb{E}}\boldsymbol{\theta}_l = \begin{bmatrix} \theta_l \\ \phi_l \end{bmatrix}$	${}^{\mathbb{H}}\boldsymbol{\theta}_l$ describes the position of the load, relative to the helicopter COM seen in the earth-fixed frame by the angles $\{\theta_l \ \phi_l\}$. Note that $\boldsymbol{\theta}_l$ is not equivalent to θ_l .

Table 7.1: Table explaining the naming and grouping of the 9 DOFs.

Figure 7.2 on the facing page shows a 3D illustration of the helicopter and slung load, where the helicopter is illustrated by a box and the load by a sphere. The figure shows in detail how the load angles are defined, in relation to the earth fixed frame, and the definition of the position vector used, which will later be used for determining the loads position relative to the helicopter.

7.1.1 State variables

The state variables are composed of the position and angles of the helicopter and angles of the load in \mathbb{E} , as shown in (7.1), and the corresponding linear and angular velocities in \mathbb{H} and \mathbb{L} , as shown in (7.2).

$$\mathbf{q} = \begin{bmatrix} {}^{\mathbb{E}}\mathbf{p}_h \\ {}^{\mathbb{E}}\boldsymbol{\theta}_h \\ {}^{\mathbb{E}}\boldsymbol{\theta}_l \end{bmatrix} \quad (7.1)$$

\mathbf{q} Position vector of the full system.

change of the helicopter position in the earth fixed frame, and the velocity is given in the helicopter frame, so the derivative has to be rotated. Similar transformations are needed for the other derivatives. This mapping is done by the following block diagonal matrix:

$$\mathbf{T}_b = \begin{bmatrix} {}^{\mathbb{H}}_{\mathbb{E}}\mathbf{R} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{J}_{\mathbb{H}} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{J}_{\mathbb{L}} \end{bmatrix} \quad (7.4)$$

${}^{\mathbb{H}}_{\mathbb{E}}\mathbf{R}$	Rotation matrix expressing the rotation from \mathbb{E} to \mathbb{H} shown in (7.6).
$\mathbf{J}_{\mathbb{H}}$	Matrix expressing the transformation from the helicopter Euler angles to it's corresponding angular velocities. This matrix is shown in (7.7).
$\mathbf{J}_{\mathbb{L}}$	Matrix expressing the transformation from the load Euler angles to it's corresponding angular velocities. This matrix is shown in (7.8).

The rotation matrix is obtained by multiplying the three individual rotation matrices, as shown in (7.6):

$$\begin{aligned} {}^{\mathbb{H}}_{\mathbb{E}}\mathbf{R} &= \mathbf{R}_x(\theta_h) \mathbf{R}_y(\phi_h) \mathbf{R}_z(\psi_h) \\ &= \begin{bmatrix} c(\phi_h) c(\psi_h) & c(\phi_h) s(\psi_h) & -s(\phi_h) \\ c(\psi_h) s(\phi_h) s(\theta_h) - c(\theta_h) s(\psi_h) & c(\psi_h) c(\theta_h) + s(\phi_h) s(\psi_h) s(\theta_h) & c(\phi_h) s(\theta_h) \\ s(\psi_h) s(\theta_h) + c(\psi_h) c(\theta_h) s(\phi_h) & c(\theta_h) s(\phi_h) s(\psi_h) - c(\psi_h) s(\theta_h) & c(\phi_h) c(\theta_h) \end{bmatrix} \end{aligned} \quad (7.5)$$

To save space, in equation (7.6) a short notation for cosine and sine is used; namely $\cos(x) = c(x)$ and $\sin(x) = s(x)$.

The transformation from Euler rates to body angular velocities is slightly more complicated, since the rotations are consecutive, some of the rotation axis are given in intermediate frames, and have to be rotated into the body frame individually. Since the rotation is in the order z-y-x (rotation around the z-axis first) the first angle rate $\dot{\theta}_h$ is already given in the helicopter frame. The second rate $\dot{\phi}_h$ is given in the intermediate frame that is before the last rotation, hence it has to be rotated by the last angle. The first rotation is given in the frame before, and thus must be rotated by the second angle and then the third angle. This is done in (7.6).

$$\boldsymbol{\omega}_h = \begin{bmatrix} \dot{\theta}_h \\ 0 \\ 0 \end{bmatrix} + \mathbf{R}_x(\theta_h) \begin{bmatrix} 0 \\ \dot{\phi}_h \\ 0 \end{bmatrix} + \mathbf{R}_x(\theta_h) \mathbf{R}_y(\phi_h) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi}_h \end{bmatrix} \quad (7.6)$$

Combining the individual rotations into $\mathbf{J}_{\mathbb{H}}$, looks as shown in (7.7).

$$\mathbf{J}_{\mathbb{H}} = \begin{bmatrix} 1 & 0 & -\sin(\phi_h) \\ 0 & \cos(\theta_h) & \cos(\phi_h) \sin(\theta_h) \\ 0 & -\sin(\theta_h) & \cos(\phi_h) \cos(\theta_h) \end{bmatrix} \quad (7.7)$$

Note that the determinant $|\mathbf{J}_{\mathbb{H}}| = \cos(\phi_h)$, which makes $\mathbf{J}_{\mathbb{H}}$ singular for $\phi_h = \pm\pi/2$.

The matrix $\mathbf{J}_{\mathbb{L}}$ can be constructed following the exact same procedure, which will result in an identical matrix:

$$\mathbf{J}_{\mathbb{L}} = \begin{bmatrix} 1 & 0 & -\sin(\phi_l) \\ 0 & \cos(\theta_l) & \cos(\phi_l) \sin(\theta_l) \\ 0 & -\sin(\theta_l) & \cos(\phi_l) \cos(\theta_l) \end{bmatrix} \quad (7.8)$$

Again note the singularity at $\phi_l = \pm\pi/2$.

Having defined the elements in \mathbf{T}_b , the differential equations describing the position and angle states can be written as:

$$\dot{\mathbf{q}} = \mathbf{T}_b^{-1} \dot{\mathbf{q}}_b \quad (7.9)$$

Differential equations describing $\dot{\mathbf{q}}_b$ are derived in the following section by considering Euler's laws of motion for a rigid body.

7.2 Conservation of Momentum

Newtons second law states that the rate of change in linear momentum, of a particle is equal to the sum of forces acting on it. Leonhard Euler expanded this to rigid bodies, by means of integration of particles. Based on this, he formulated Euler's laws of motion of a rigid body [12]. These laws are presented in (7.10) and (7.11).

Euler's first law states that the rate of change in linear momentum of a body is equal to the total force applied, as described by (7.10).

$$\frac{d(m\mathbf{v})}{dt} = \mathbf{F} \quad (7.10)$$

m	Mass of the body.	kg
\mathbf{v}	Velocity of the centre of mass.	m/s
\mathbf{F}	Total force applied.	N

Euler's second law states that the rate of change in angular momentum of a body, about an arbitrary point p is equal to the total torque about that point, as described by (7.11).

$$\frac{d(\mathbf{I}\boldsymbol{\omega})}{dt} = \boldsymbol{\tau} \quad (7.11)$$

\mathbf{I}	Moment of inertia about point p .	kg m ²
$\boldsymbol{\omega}$	Angular velocity about point p .	rad/s
$\boldsymbol{\tau}$	Total torque about point p .	N m

Figure 7.3 on the next page shows the free body diagram of the of the bodies, shown in figure 7.2 on page 47.

The free body diagram introduces the internal string force, denoted by \mathbf{F}_s which is the force that binds the bodies together, and allows the bodies to be treated individually. The First body treated is the helicopter body, on the figure represented by a box.

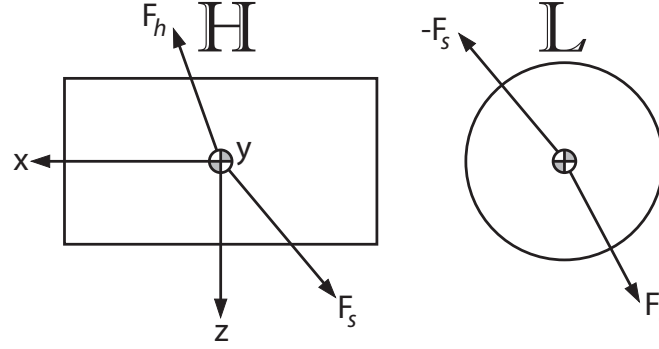


Figure 7.3: Free-body diagram of the bodies involved in the rigid body model, showing the forces acting on the load and helicopter separately.

7.2.1 Linear momentum of the helicopter

Applying (7.10) on the helicopter yields the following equation:

$$m_h {}^{\mathbb{E}}\mathbf{a}_h = {}^{\mathbb{E}}\mathbf{F}_h + {}^{\mathbb{E}}\mathbf{F}_s \quad (7.12)$$

\mathbf{a}_h	Linear acceleration of the helicopter.	N
\mathbf{F}_h	Sum of the external forces acting on the helicopter, which are derived in [7] and listed in table 5.1 on page 25.	N
\mathbf{F}_s	Reaction force originating from the tension on the line.	N
m_h	Mass of the helicopter.	kg

\mathbf{F}_s is unknown, but can be described by applying the same method on the load body, and isolating the force. This is done in (7.13).

$${}^{\mathbb{E}}\mathbf{F}_s = {}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{a}_l \quad (7.13)$$

\mathbf{a}_l	Linear acceleration of the load	m/s ²
\mathbf{F}_l	Sum of the external forces acting on the load. Expressions for these are derived in chapter 6 on page 29.	N
m_l	Mass of the load.	kg

Combining (7.12) and (7.13), yields an expression for the acceleration of the helicopter, which is dependent of the acceleration of the load, as shown in (7.14).

$$m_h {}^{\mathbb{E}}\mathbf{a}_h = {}^{\mathbb{E}}\mathbf{F}_h + {}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{a}_l \quad (7.14)$$

The acceleration of the load can be found by differentiating the following expression for the position of the load, which is given in (7.15)

$${}^{\mathbb{E}}\mathbf{p}_{HL} = {}^{\mathbb{E}}\mathbf{R}^{\mathbb{L}} \mathbf{p}_{HL} \quad (7.15)$$

\mathbf{p}_{HL}	Position vector from \mathbb{H} to \mathbb{L} . \mathbf{p}_{HL} is the position of the load relative to the helicopter COM.	m
-------------------	---	---

Differentiating yields:

$$\mathbb{E}\mathbf{v}_l = \mathbb{E}\mathbf{R} \left(\mathbb{L}\boldsymbol{\omega}_l \times \mathbb{L}\mathbf{p}_{HL} \right) + \mathbb{E}\mathbf{v}_h \quad (7.16)$$

\mathbf{v}_l	Velocity vector of the load relative to the helicopter COM.	m/s
----------------	---	-----

Further differentiation yields:

$$\mathbb{E}\mathbf{a}_l = \mathbb{E}\mathbf{R} \left(\mathbb{L}\dot{\boldsymbol{\omega}}_l \times \mathbb{L}\mathbf{p}_{HL} + \mathbb{L}\boldsymbol{\omega}_l \times \left(\mathbb{L}\boldsymbol{\omega}_l \times \mathbb{L}\mathbf{p}_{HL} \right) \right) + \mathbb{E}\mathbf{a}_h \quad (7.17)$$

Since the position of the load in the load fixed frame is constant, there is no Coriolis term in the equation. For more information on movement in rotating and translating coordinate systems, see Appendix C on page 181. Combining (7.14) and (7.17) yields the following expression of the helicopter acceleration:

$$(m_h + m_l)\mathbb{E}\mathbf{a}_h = \mathbb{E}\mathbf{F}_h + \mathbb{E}\mathbf{F}_l - m_l\mathbb{E}\mathbf{R} \left(\mathbb{L}\dot{\boldsymbol{\omega}}_l \times \mathbb{L}\mathbf{p}_{HL} + \mathbb{L}\boldsymbol{\omega}_l \times \left(\mathbb{L}\boldsymbol{\omega}_l \times \mathbb{L}\mathbf{p}_{HL} \right) \right) \quad (7.18)$$

The needed equation for the state vector is $\mathbb{H}\dot{\mathbf{v}}_h$, which can be found by the following equation:

$$\mathbb{H}\dot{\mathbf{v}}_h = \mathbb{H}\mathbf{R} \mathbb{E}\mathbf{a}_h - \mathbb{H}\boldsymbol{\omega}_h \times \mathbb{H}\mathbf{v}_h \quad (7.19)$$

The following section will derive an expression for the rate of change of the angular velocity.

7.2.2 Angular momentum of the slung load

This section will derive an expression for the load angular acceleration, and will use (7.11), as the fundamental theorem. (7.20) shows (7.11) directly applied to the load angles.

$$\mathbb{E}\dot{\mathbf{L}}_l = \mathbb{E}\boldsymbol{\tau}_l \quad (7.20)$$

\mathbf{L}_l	Angular momentum of the load, which equals $\mathbf{L}_l = (\mathbf{I}_l\dot{\boldsymbol{\omega}}_l)$.	N m s
\mathbf{I}_l	Moment of inertia of the load with respect to the point H	kg m ²
$\boldsymbol{\tau}_l$	Torque applied to the load in the point H	N m/s ²

$\mathbb{E}\mathbf{I}_l$ is however time dependent. This can be overcome by perceiving the angular momentum as a vector in the rotating frame \mathbb{L} , where the moment of inertia is constant. The time derivative of the angular momentum can then be described as:

$$\mathbb{E}\dot{\mathbf{L}}_l = \mathbb{E}\mathbf{R} \left(\mathbb{L}\dot{\mathbf{L}}_l + \mathbb{L}\boldsymbol{\omega}_l \times \mathbb{L}\mathbf{L}_l \right) \quad (7.21)$$

Now that the moment of inertia matrix is constant, the expression can be written as:

$$\mathbb{E}\dot{\mathbf{L}}_l = \mathbb{E}\mathbf{R} \left(\mathbb{L}\mathbf{I}_l \mathbb{L}\dot{\boldsymbol{\omega}}_l + \mathbb{L}\boldsymbol{\omega}_l \times \mathbb{L}\mathbf{I}_l \mathbb{L}\boldsymbol{\omega}_l \right) \quad (7.22)$$

where the moment of inertia matrix is given as $\mathbb{L}\mathbf{I}_l = m_l l^2 \mathbf{E}_3$, where \mathbf{E}_3 is the three by three identity matrix, and l is the length of the wire. That the last diagonal component is equal to the others is purely to make the calculations possible, since in reality it should be zero, but such a matrix is not possible to invert. Since the third component

of the angle is not important the magnitude of it is unimportant. Now reinserting (7.22) into (7.20) and isolating for ${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l$ yields:

$${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l = {}^{\mathbb{L}}\mathbf{I}_l^{-1} \left({}^{\mathbb{L}}\boldsymbol{\tau}_l - {}^{\mathbb{L}}\boldsymbol{\omega}_l \times {}^{\mathbb{L}}\mathbf{I}_l {}^{\mathbb{L}}\boldsymbol{\omega}_l \right) \quad (7.23)$$

The ${}^{\mathbb{L}}\boldsymbol{\omega}_l \times {}^{\mathbb{L}}\mathbf{I}_l {}^{\mathbb{L}}\boldsymbol{\omega}_l$ term is always zero, since evaluating the cross product of a vector and a scaling of it self is always zero.

The torque applied on the load seen in point H is given as:

$${}^{\mathbb{L}}\boldsymbol{\tau}_l = {}^{\mathbb{L}}\mathbf{p}_{HL} \times {}^{\mathbb{L}}\mathbf{R} \left({}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{a}_l \right) \quad (7.24)$$

${}^{\mathbb{E}}\mathbf{a}_l$ is given by (7.14). Combining (7.23), (7.24), (7.14) and (7.17), provides an expression of the rate of change of the angular velocity:

$${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l = {}^{\mathbb{L}}\mathbf{I}_l^{-1} \left({}^{\mathbb{L}}\mathbf{p}_{HL} \times {}^{\mathbb{L}}\mathbf{R} \frac{m_h {}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{F}_h - m_h m_l {}^{\mathbb{L}}\mathbf{R} ({}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l \times {}^{\mathbb{L}}\mathbf{p}_{HL} + {}^{\mathbb{L}}\boldsymbol{\omega}_l \times ({}^{\mathbb{L}}\boldsymbol{\omega}_l \times {}^{\mathbb{L}}\mathbf{p}_{HL}))}{m_h + m_l} \right) \quad (7.25)$$

${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l$ is however present on both sides of the equality sign. By introducing the following skew-symmetric matrix, to replace the $\mathbf{p}_{HL} \times$ operation:

$$[{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} = \begin{bmatrix} 0 & -\mathbf{p}_{HL,3} & \mathbf{p}_{HL,2} \\ \mathbf{p}_{HL,3} & 0 & -\mathbf{p}_{HL,1} \\ -\mathbf{p}_{HL,2} & \mathbf{p}_{HL,1} & 0 \end{bmatrix} \quad (7.26)$$

Substituting this matrix yields the following expression:

$${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l = {}^{\mathbb{L}}\mathbf{I}_l^{-1} [{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} \frac{{}^{\mathbb{L}}\mathbf{R} (m_h {}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{F}_h) + m_h m_l \left([{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} {}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l + {}^{\mathbb{L}}\boldsymbol{\omega}_l \times ([{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} {}^{\mathbb{L}}\boldsymbol{\omega}_l) \right)}{m_h + m_l} \quad (7.27)$$

Note that $-m_l [{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} [{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times}$ by definition equals the inertia matrix ${}^{\mathbb{L}}\mathbf{I}_l$. Therefore (7.27) can be expressed as:

$${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l = \frac{1}{m_l + 2m_h} {}^{\mathbb{L}}\mathbf{I}_l^{-1} [{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} \left({}^{\mathbb{L}}\mathbf{R} (m_h {}^{\mathbb{E}}\mathbf{F}_l - m_l {}^{\mathbb{E}}\mathbf{F}_h) + m_h m_l {}^{\mathbb{L}}\boldsymbol{\omega}_l \times [{}^{\mathbb{L}}\mathbf{p}_{HL}]_{\times} {}^{\mathbb{L}}\boldsymbol{\omega}_l \right) \quad (7.28)$$

This expression for the rate of change of the angular velocity of the load can then be inserted into the expression for the acceleration of the helicopter. The last equation needed is the equation for the rate of change of the helicopter angular velocity, which will be derived in the following section.

7.2.3 Angular momentum of the helicopter

The rate of change of the helicopter angular velocity is given by an equation similar to (7.23), only here the torque is given as an input to the system. The inertia matrix for the helicopter ${}^{\mathbb{H}}\mathbf{I}_h$ is constant and is derived in [7].

$${}^{\mathbb{H}}\dot{\boldsymbol{\omega}}_h = {}^{\mathbb{H}}\mathbf{I}_h^{-1} \left({}^{\mathbb{H}}\boldsymbol{\tau}_h - {}^{\mathbb{H}}\boldsymbol{\omega}_h \times {}^{\mathbb{H}}\mathbf{I}_h {}^{\mathbb{H}}\boldsymbol{\omega}_h \right) \quad (7.29)$$

\mathbf{I}_h	Moment of inertia of the helicopter with respect to the point H	kg m ²
$\boldsymbol{\tau}_h$	Sum of the external torques acting on the helicopter, which is derived in [7] and listed in table 5.2 on page 26.	N m

The following section will summarise the equations describing the states of the system, which has been derived in this chapter.

7.3 Equation Summary

So far in this chapter, differential equations for describing the states of the helicopter and slung load has been derived. This section will summarise these, beginning with the expressions for the positions derivatives.

$$\begin{bmatrix} {}^{\mathbb{E}}\dot{\mathbf{p}}_h \\ {}^{\mathbb{E}}\dot{\boldsymbol{\theta}}_h \\ {}^{\mathbb{E}}\dot{\boldsymbol{\theta}}_l \end{bmatrix} = \mathbf{T}_b^{-1} \begin{bmatrix} {}^{\mathbb{H}}\mathbf{v}_h \\ {}^{\mathbb{H}}\boldsymbol{\omega}_h \\ {}^{\mathbb{L}}\boldsymbol{\omega}_l \end{bmatrix} \quad (7.30)$$

\mathbf{p}_h	Position of the helicopter.	m
$\boldsymbol{\theta}_h$	Euler angles of the helicopter.	rad
$\boldsymbol{\theta}_l$	Angles of the load relative to the helicopter.	rad
\mathbf{T}_b	Transformation matrix mapping position and angle derivatives to body velocities. Described in detail in (7.4)	
\mathbf{v}_h	Velocity of the helicopter.	m/s
$\boldsymbol{\omega}_h$	Angular velocity of the helicopter.	rad/s
$\boldsymbol{\omega}_l$	Angular velocity of the load relative to the helicopter.	rad/s

Following the differential equations describing the velocities will be shown.

$${}^{\mathbb{H}}\dot{\mathbf{v}}_h = \frac{{}^{\mathbb{H}}\mathbf{F}_h + {}^{\mathbb{H}}\mathbf{F}_l - m_l {}^{\mathbb{H}}\mathbf{R}({}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l \times {}^{\mathbb{L}}\mathbf{p}_{HL} + {}^{\mathbb{L}}\boldsymbol{\omega}_l \times ({}^{\mathbb{L}}\boldsymbol{\omega}_l \times {}^{\mathbb{L}}\mathbf{p}_{HL}))}{m_h + m_l} - {}^{\mathbb{H}}\boldsymbol{\omega}_h \times {}^{\mathbb{H}}\mathbf{v}_h \quad (7.31)$$

$${}^{\mathbb{H}}\dot{\boldsymbol{\omega}}_h = {}^{\mathbb{H}}\mathbf{I}_h^{-1} \left({}^{\mathbb{H}}\boldsymbol{\tau}_h - {}^{\mathbb{H}}\boldsymbol{\omega}_h \times {}^{\mathbb{H}}\mathbf{I}_h {}^{\mathbb{H}}\boldsymbol{\omega}_h \right) \quad (7.32)$$

$${}^{\mathbb{L}}\dot{\boldsymbol{\omega}}_l = \frac{{}^{\mathbb{L}}\mathbf{I}_l^{-1}}{m_l + 2m_h} \left({}^{\mathbb{L}}\mathbf{p}_{HL} \times \frac{m_h {}^{\mathbb{L}}\mathbf{F}_l - m_l {}^{\mathbb{L}}\mathbf{F}_h + m_h m_l {}^{\mathbb{L}}\boldsymbol{\omega}_l \times ({}^{\mathbb{L}}\boldsymbol{\omega}_l \times {}^{\mathbb{L}}\mathbf{p}_{HL})}{m_h + m_l} \right) \quad (7.33)$$

\mathbf{F}_h	External force acting on the helicopter.	N
\mathbf{F}_l	External force acting on the load.	N
$\boldsymbol{\tau}_h$	External torque acting on the helicopter COM.	N m
\mathbf{p}_{HL}	Position of the load relative to the helicopter.	m
\mathbf{I}_h	Moment of inertia matrix of the helicopter, given in the helicopter COM.	kg m ²
\mathbf{I}_l	Moment of inertia matrix of the load, given in the helicopter COM.	kg m ²

This summary concludes this chapter, in the following chapter a qualitative analysis of the results found in this and the preceding chapter will be presented.

Model Simulation and Verification

Here the simulation and verification of the slung load and rigid body model is presented. Section 8.1 presents the developed MATLAB script. Then, a comparison between simulation and experimentation will be used to verify aspects of the model.

8.1 Slung Load Simulation

To be able to verify the slung load model, a program containing all of the equations of the model had to be produced. Here a simulation is presented that include the equations provided in chapter 6 starting on page 29.

A general presentation on the structure of the program will be followed by the presentation of the results. The main script and all the functions are available on the DVD:



/Simulation/Buoy/main.m
/Simulation/Buoy/*

8.1.1 Matlab Script

The aim of the MATLAB script is to compute the state of the buoy over time. The state in this context being the position and velocity of the buoy. At each time step, the algorithm computes the forces acting on the buoy at the given time. These forces result in an acceleration, which is then integrated once to get velocity and again to obtain position.

The first simulation is of a simple case where the buoy is held in a position above its steady state, and then released to observe the dynamics of the buoy. The expected result is that the load oscillate up and down in the water, with no movement in the horizontal plane. The drag force should provide a damping of the oscillations, which should converge to the gravity-buoyancy force equilibrium. The vertical position of the buoy is shown in figure 8.1.

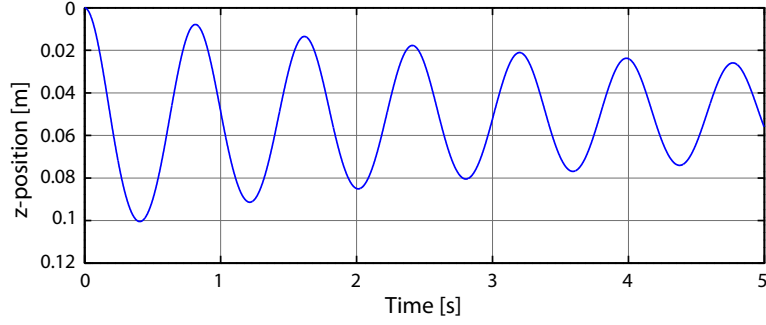


Figure 8.1: z-position of the buoy over a 5 second simulation.

The lift and drag forces are the source of damping in the system and is therefore important. Since the lift force in this case is zero, only the drag force is illustrated in figure 8.2. As expected the drag force is large in the beginning, and changes sign with the velocity of the buoy.

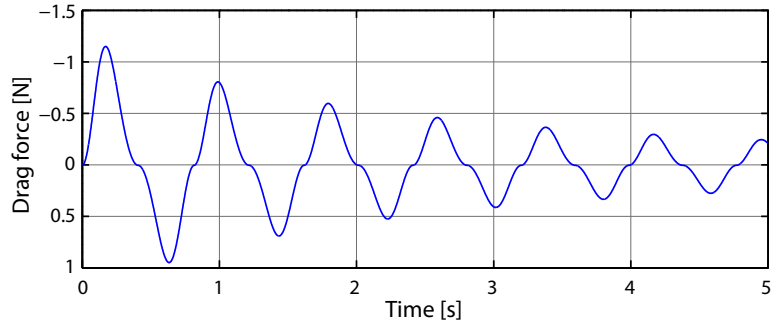


Figure 8.2: Graphical representation of the drag force magnitude over time.

8.2 Vertical Movement Experiment

This experiment is designed to give an indication of whether the derived slung load model forces can be compared to those acting on a real buoy, in a small scale setup where the buoy is placed in a tub with water.

The load is released at a position, different from the steady state position. In order to do this, the load is left in the tank until it reaches steady state, then lifted about 1.5 – 3.0 cm and released. The resulting motion is then observed.

The experiments are conducted in the AAU motion-tracking laboratory, using the *VI-CON MX UTRANET HD* system with *Vicon Tracker 1.2* software. Data is saved and formatted in MATLAB using *LoadViconDataStreamSDK()*. The setup is shown in figure 8.3.

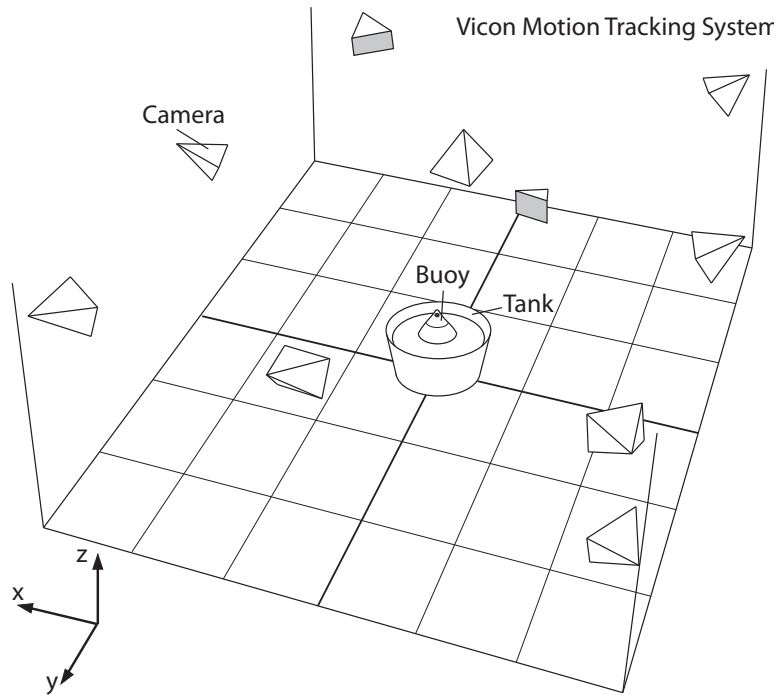


Figure 8.3: Illustration of the motion tracking setup.

The data provided by the motion-tracking system have been formatted to fit the simulation comparison. Meaning that zero-values before and after the experiment have been removed, and the altitude position of the load has been corrected so the altitude value represent the altitude of the buoy Center of Mass, this is done since the only way to track the buoy in water is to attach tracking balls on the part of the buoy above water.

A simulation of the experimental setup, using the derived slung load model, is then run for comparison between the two. The results of the experiment and the results of the corresponding simulation are presented in figure 8.4 on the following page.

The observed differences between the two curves in figure 8.4 include a difference in the frequency, and a more noticeable difference in the damping. The comparrison script can be found on the DVD along with the rest of the buoy simulation scripts



/Simulation/Buoy/check_lab.m

With the small scale experiment setup being the load in a tub with water, some undesirable effects are introduced to the experiment. These effects include the water reflecting on the sides of the tub, creating waves. A second effect is the water surface moving up and down, from the water displaced by the loads movement, due to the small surface area in the tub. Despite these effects the experiment can still be used to verify the if the slung load model is behaving in the same way as the experiment.

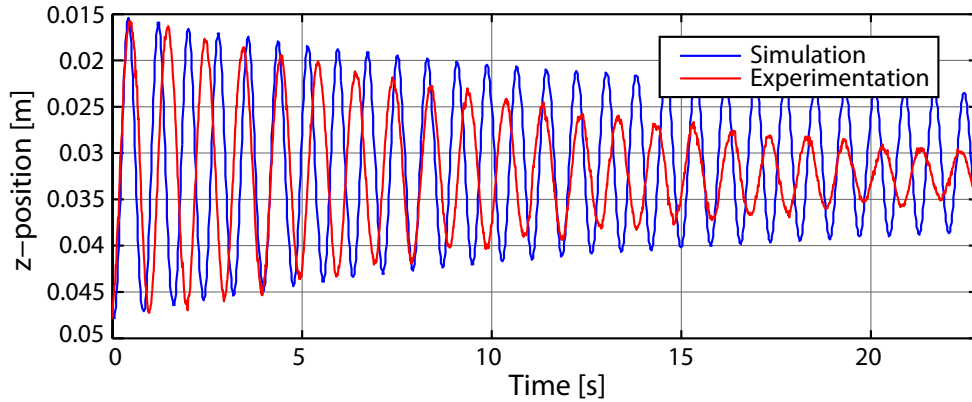


Figure 8.4: Altitude position of the load from the simulation and the experiment.

The rising and falling water level increase the relative velocity of the buoy in relation to the water, increasing the drag force on the buoy, however this effect might be negligible.

In the simulation results, it appears that the damping is too small. The damping effects are explained by the model through the drag and lift forces as explained in section 6.3 on page 33. The steady state value appear to be the same for both experimentation and simulation, which means the static states are the same in both cases. The frequency of the dynamics are within the same range, but with the simulation results having a slightly higher frequency.

8.3 External Force Simulations

This section will introduce some simulation results that have not been validated by experimentation. The aim of these simulations is to indicate if the behaviour of the buoy appear correct.

Pull on x-axis

In the first test, the load is started in steady state. Then, an external force is applied in the x-direction. The x- and z-positions of the load are represented in figure 8.5. The simulation start with no external force at time $t = 0$ s, then a force of 10 N is applied in the positive x-direction at $t = 3$ s, and finally the load is released again at $t = 7$ s.

The y-position is not shown in figure 8.5 because there is no motion on y-direction. The results correspond to the expected behaviour of the load: when a force is applied on it, it starts to move in the direction of the force and due to lift force the load goes a bit deeper in the water.

Pull on xz-plan

This test will apply a force in the x- and z-direction, as a helicopter would do while dragging it. For the simulation, the x-component of the external force is 10 N and the

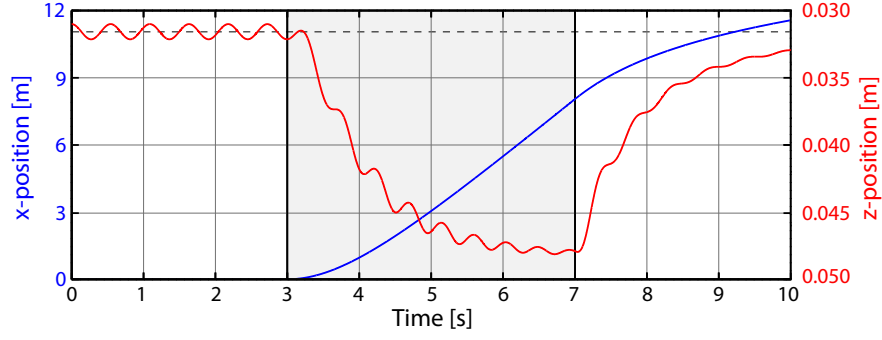


Figure 8.5: Plot of the load position in x- and z-directions over time, given a force of 10 Newtons in the interval between three and seven seconds.

z-component is 20 N. The external force is applied from $t = 3$ s to $t = 7$ s. The results are presented in figure 8.6.

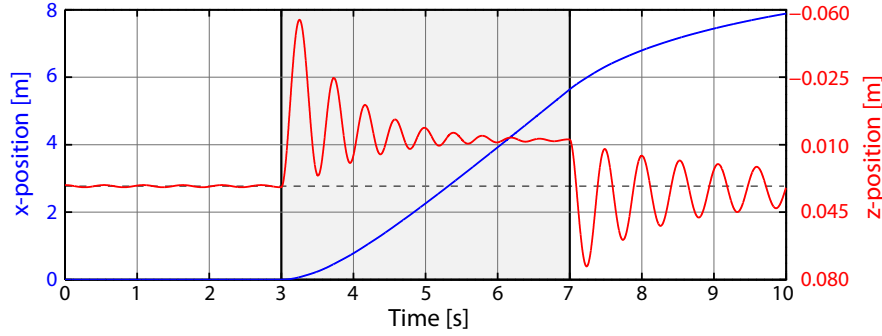


Figure 8.6: Plot representing the position in x and z direction over time, for a 2 components external force.

The results in figure 8.6 are the one expected. At $t = 3$ s, the load goes upward because of the z-component of the external force. Then, the load move in the x- and z-directions. And at $t = 7$ s, the external force is removed, so the load goes back towards steady state.

8.4 Conclusion of Slung Load Model Simulations

The simulations of the slung loads behavior when released or with an external force applied, shows the expected tendencies in the dynamics. The experiment, however, shows some differences. The differences observed are partially explained by experimental setup. The analysis and comparison highlight the fact that frequency and static state are reasonable, and the damping is too low. Considering the frequency, damping and steady state, the model is able to obtain close to equal dynamics in frequency and steady state value. The model is then deemed adequate for further use in this work.

8.5 Rigid Body Model Simulations

As with the slung load model, a verification of sorts is wanted for the rigid body modelling part of the project, to evaluate whether the results from the rigid body modelling chapter is usable for further use in the control part of this work. Actual verification does however require real data for comparison, which is not available for this work. Instead the simulations performed in this chapter, is set up to see whether different actions make the model perform as expected.

The simulations has been split up into two categories, one with simulations where the model is given an input, and the response is observed, these inputs are torque on the helicopter and force applied on either the helicopter or the slung load. The other category is where the simulations start with the model set at some initial conditions, and the response is observed here as well, these tests include the load set at an angle or with an initial angular velocity. To test the capabilities of the model, some simulations have been set up, contained in both categories.

8.5.1 Force Applied onto the Load

The first input simulation is force applied on the load in the xy-plane with the components shown in figure 8.7 and the z-component of the load force constant in the simulations set to be equal to the gravity force on the load.

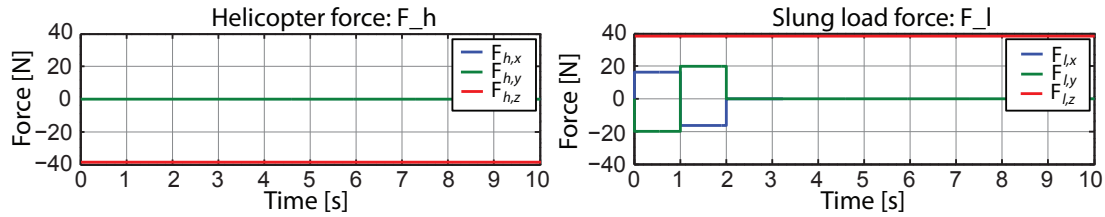


Figure 8.7: A graph showing the force components of the force applied on the helicopter and slung load.

The force applied on the load is set to last for 1 second, followed by an equal and opposite force in 1 second, with 8 seconds left of the simulation to observe the response. Figure 8.8 show the position of the helicopter and slung load, 8.9 the angular position, velocity and acceleration of the load together with the translatory position, velocity and acceleration of the helicopter.

The positions of the helicopter and load show how the force applied onto the load make it move and drag the helicopter behind it, when the opposite force is applied, the movement is changed to a swinging motion.

The angular position and derivatives of the helicopter is not shown since they are all equal to zero due to the modelling compromise of attaching the slung load cable in the helicopters centre of mass.

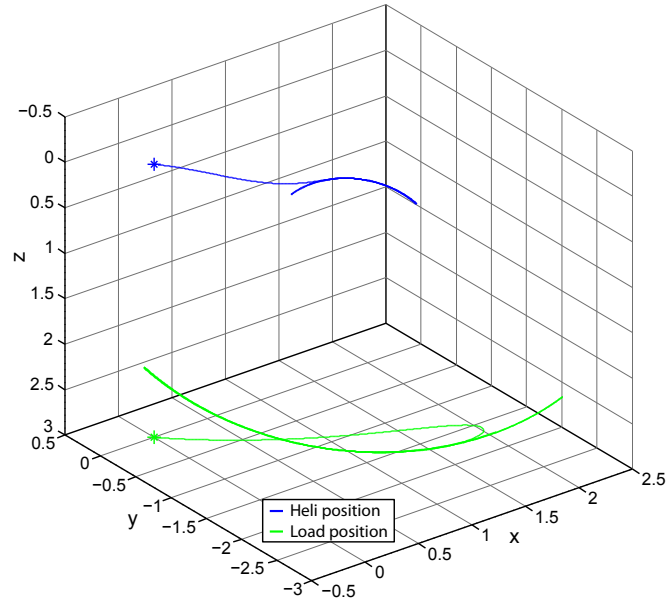


Figure 8.8: 3D plot showing the position of the helicopter and the slung load.

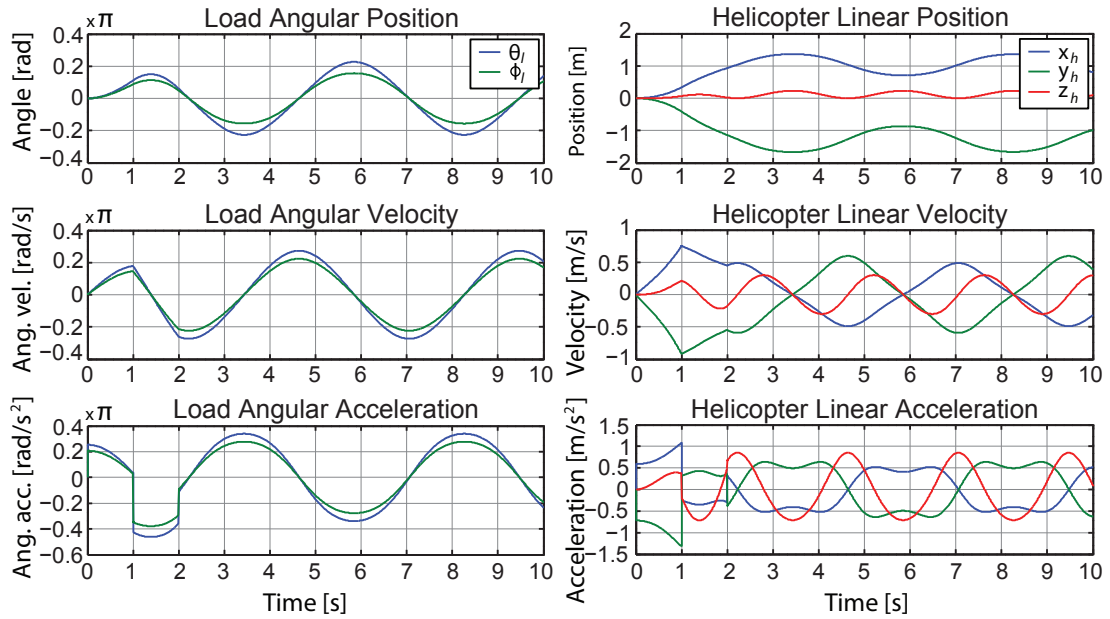


Figure 8.9: Left column show 3 plots with the loads angular position, velocity and acceleration. Note that the vertical axis in left column has been scaled with π . The right column show 3 plots containing the helicopters translatory position, velocity and acceleration.

Figure 8.9 show that even through an opposite force is applied, the motion of the load does not stop, but is simply changed to a swinging that is continuous due to the lack of damping in the system.

8.5.2 Force Applied onto the Helicopter

This test applies a force onto the helicopter in the xy-plane, which is just like the previous test where the target of the force is the load. The force is generated as shown in figure 8.10, with a given force in 1 second, followed by an opposite force in 1 second, with 8 seconds simulation left to observe the response.

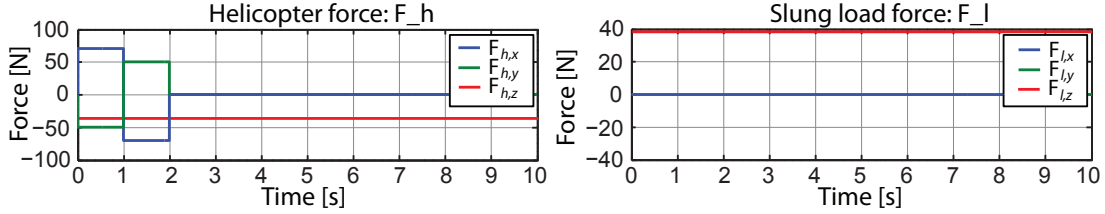


Figure 8.10: A graph showing the force components of the force applied on the helicopter and slung load.

Figure 8.11 show the position of the helicopter and slung load, and 8.12 show two columns of graphs, the left with the loads angular position and derivatives, together with the right column showing the helicopters translatory position and derivatives.

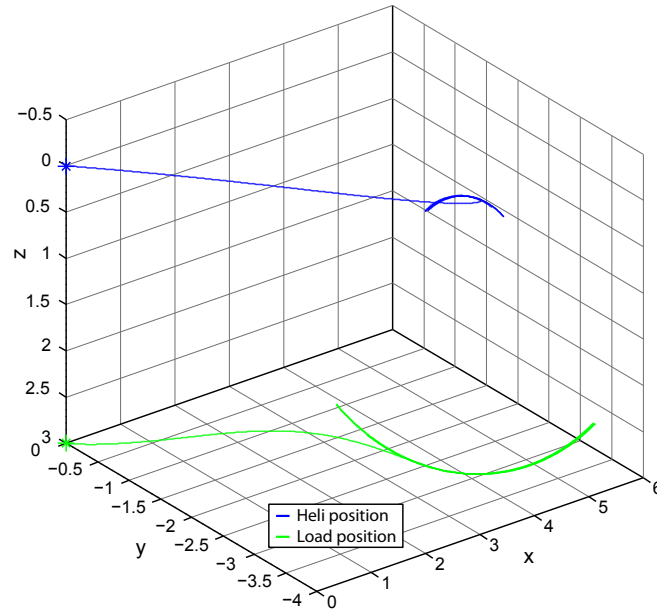


Figure 8.11: 3D plot showing the position of the helicopter and the slung load.

Figure 8.11 show that when the force is applied onto the helicopter, it start moving in a direction, dragging the load after it, when the opposite force is applied, the helicopter slows down, transforming the loads movement to a swinging motion, just like in the first test, with the difference being that it in this test is the helicopter that is slowed down.

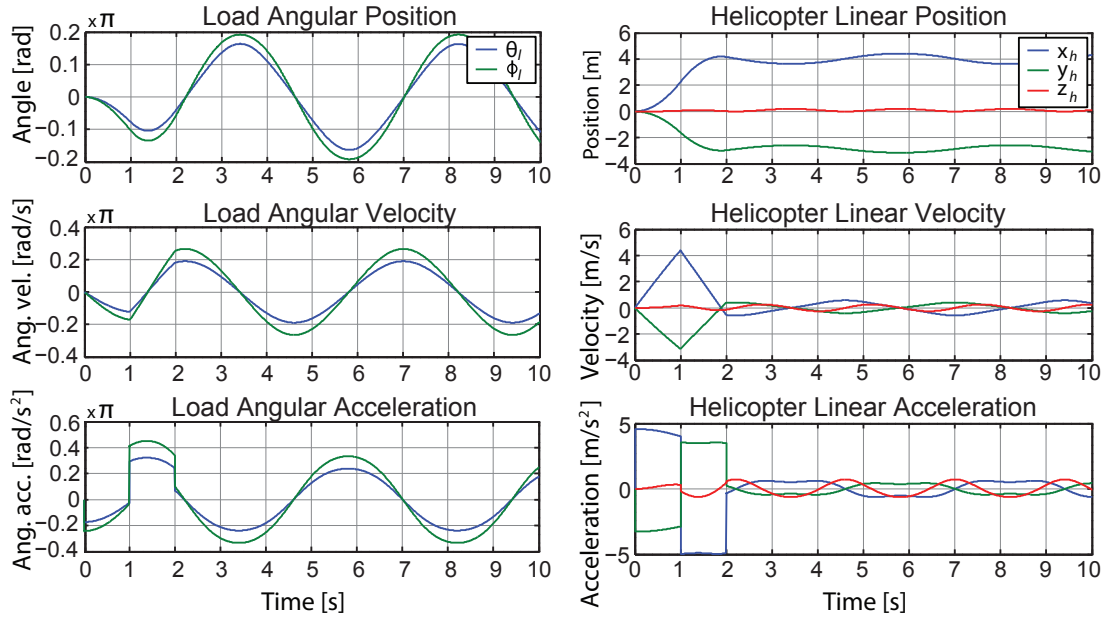


Figure 8.12: Left column show 3 plots with the loads angular position, velocity and acceleration. Note that the vertical axis in left column has been scaled with π . The right column show 3 plots containing the helicopters translatory position, velocity and acceleration.

The difference between applying a force on the load or helicopter is shown in figure 8.12 where the angular acceleration of the load now start out as negative, since the load is being dragged, and then going positive when the helicopter is slowed down by the force, which is the opposite of what is shown in figure 8.9. The angular movement of the helicopter is not shown since it is still equal to zero.

8.5.3 Torque Applied on Helicopter

The third input simulation is a torque applied onto the helicopter, around the second and third axis as shown in the last plot of figure 8.13, with the torques changing over the first 4 seconds, and is then reversed in the second half of the simulation.

The torque multiplied with the helicopters inertia around the respective axes can be observed directly through the helicopters angular acceleration in figure 8.13. The angular velocity looks as expected with constant accelerations, and the angular position is rotated to a specific point during the first 4 seconds, and then afterwards rotated back to the initial position at the end of the simulation.

The helicopter and slung loads translatory movement, and the loads angular movement have not been shown, since these are all equal to zero, due to them being uncorrelated to the helicopters angular movement, while the slung load cable is attached in the helicopters centre of mass.

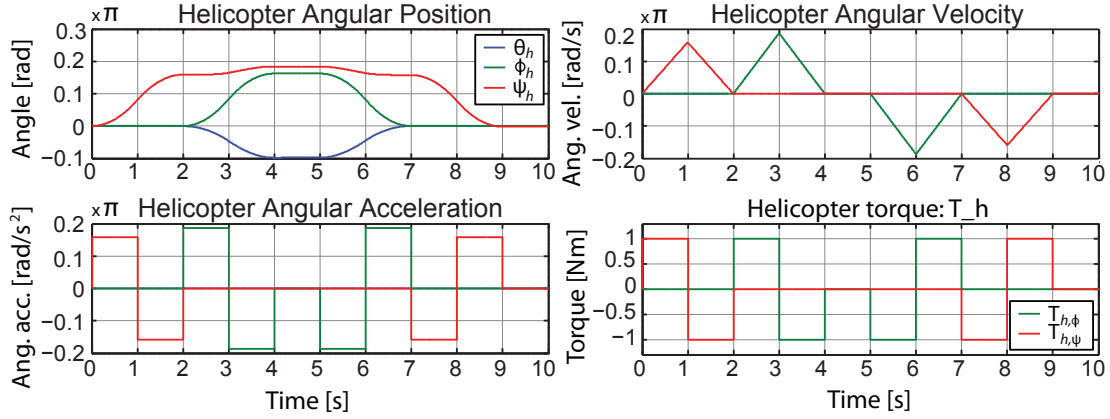


Figure 8.13: 3 plots showing the helicopters angular position, velocity and acceleration, and a final plot showing the input torque.

8.5.4 Load at an Angle

Here the slung load is set to an initial angle of $\pi/4$ rad in the θ_l and ϕ_l angles, and is then released to be observed in the 10 seconds the simulation is running. Figure 8.14 show the position of the helicopter and slung load after the load has been released. The starting position of the helicopter and load is also shown. When the load is released, it starts swinging downwards due to the gravity, as well as dragging the helicopter towards it. The helicopters lift is set to be equal to the gravity of the slung load, so when the string force is not pointing directly downwards, the helicopter rises a little. Without any disturbances on the system, the slung load swings back and forth in the same path.

Figure 8.15 show the loads angular acceleration at the bottom of the figure, since the load is released in a positive angle, the acceleration starts out negative, and the velocity goes negative due to that, bringing the slung load down towards zero angles, but since the system is without any damping, the load passes zero, and goes to a negative angle equal to the initial angle.

8.5.5 Load at an Angle, Torque on Helicopter

This simulation is a combination of the third and fourth simulation, to test whether the angular movement of the helicopter is truly uncorrelated to the movement of the load, and the helicopters translational movement.

The torque applied on the load is equal to the one shown in figure 8.13 in the third simulation, and the angle of the slung load is set to be $\pi/4$ in the first two angles, as in the fourth simulation.

The results of the simulation is not shown here, since it is exactly equal to the results shown in figures 8.13 and 8.15 from simulation 3 and 4, showing that the angular movement of the helicopter is truly uncorrelated from the translatory movement, and the slung load entirely.

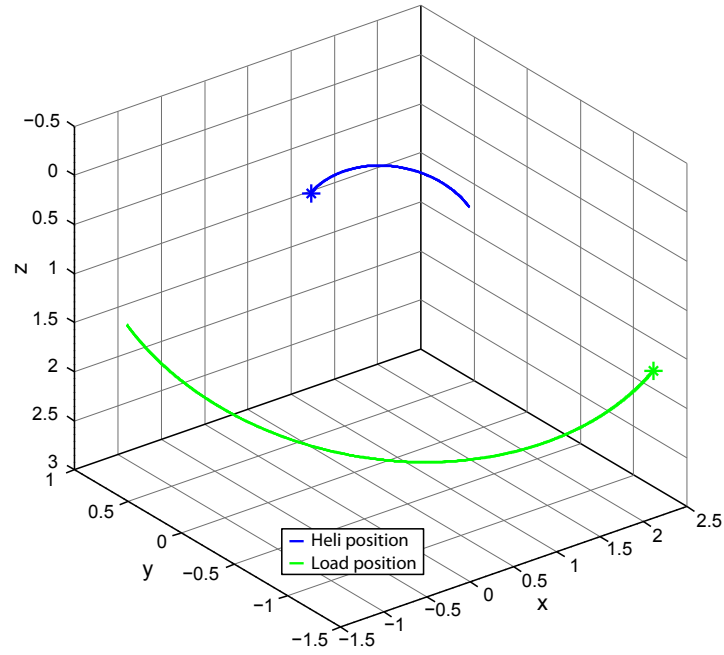


Figure 8.14: 3D plot showing the position of the helicopter and the slung load.

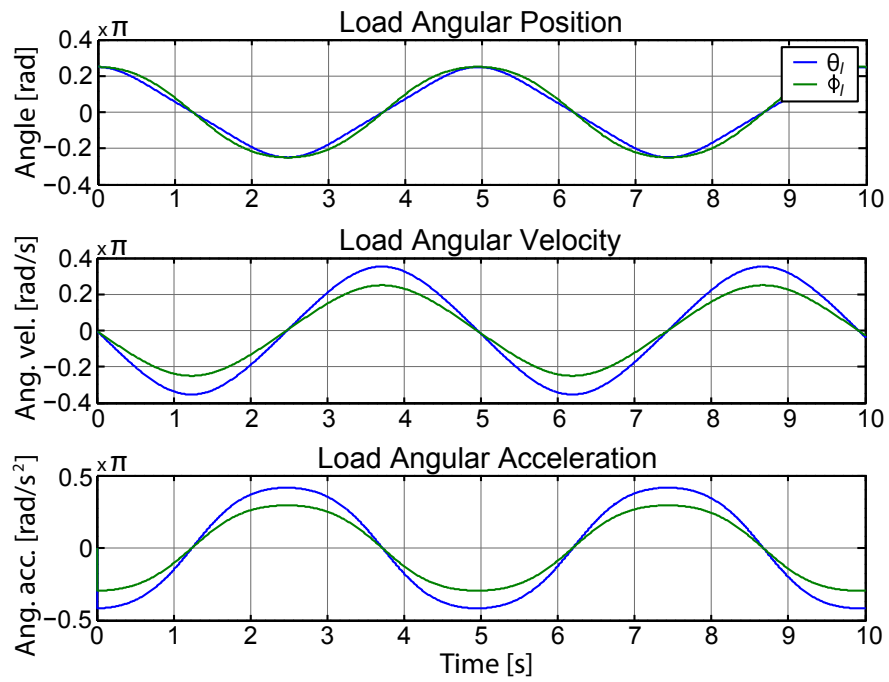


Figure 8.15: 3 plots showing the loads angular position, velocity and acceleration.

8.5.6 Load at an Angle, Force Applied onto the Helicopter

This is a combination of second and fourth simulation, this time with a force applied onto the helicopter, while having the load at an initial angle. To get a better graphical representation of the simulation in a 3D plot, the force applied onto the helicopter is only set in 1 axis, with the angle of the load set as a rotation around the same axis. The force applied onto the helicopter is set to 10 N for 1 second in the direction of the x-axis, and 0 afterwards.

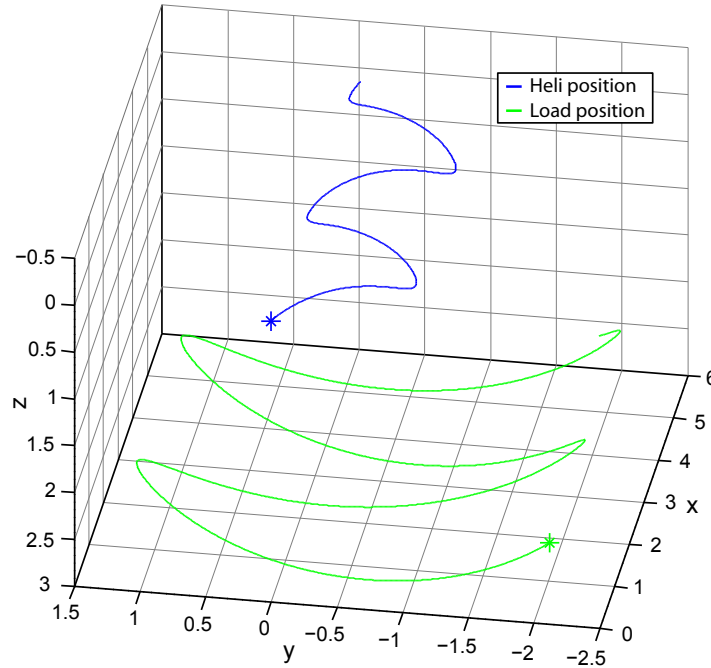


Figure 8.16: 3D plot showing the position of the helicopter and the slung load.

While subjected to the x-directional force, the helicopter start moving in that direction, while being pulled from side to side in the y-direction from the slung load swinging beneath it. In the period where the helicopter is no longer subjected to the force, the acceleration in the x-direction goes to nearly zero, while the pulls from side to side remains the same. This is all shown in figures 8.16 and 8.17

8.5.7 Load with an Initial Angular Velocity

Here the load is released with an initial angular velocity of 3rad/s around the z-axis in the earth fixed frame. The helicopter is held still, meaning the translational acceleration is set to zero, so only the dynamics of the load is observed. The load is held at an angle as well, to see the load move around the z-axis in the inertial frame.

Figure 8.18 show the position of the helicopter that is set to zero, and is only illustrated with the point where it is held, and the position of the load, rotating around the z-axis of the inertial frame. When the simulation is started, the illustration shows that the

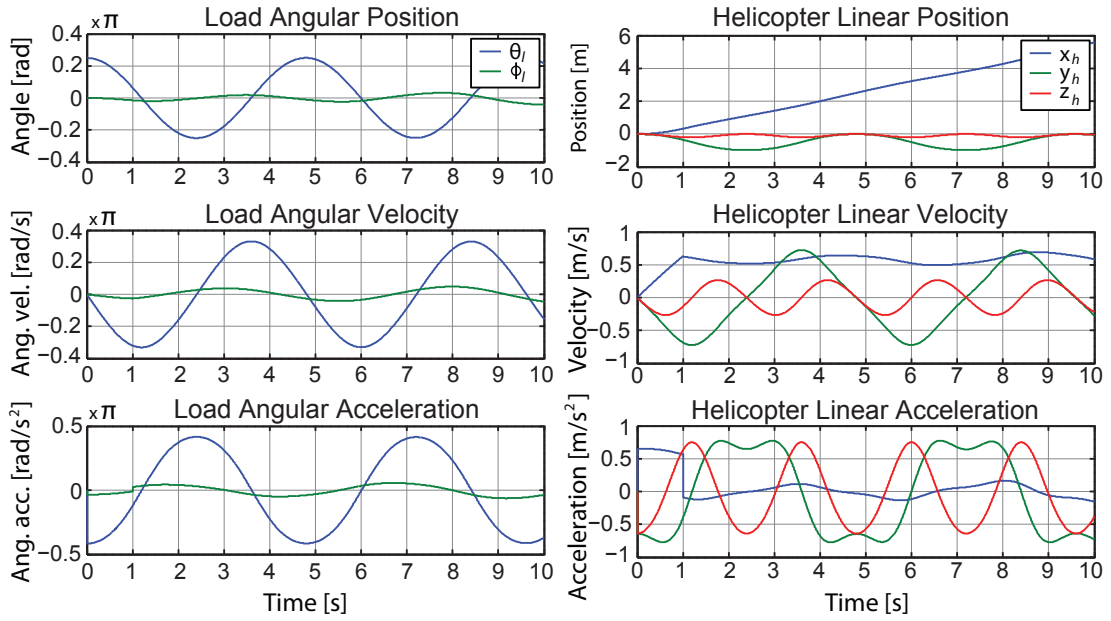


Figure 8.17: Left column show 3 plots with the loads angular position, velocity and acceleration. Note that the vertical axis in left column has been scaled with π . The right column show 3 plots containing the helicopters transatory position, velocity and acceleration.

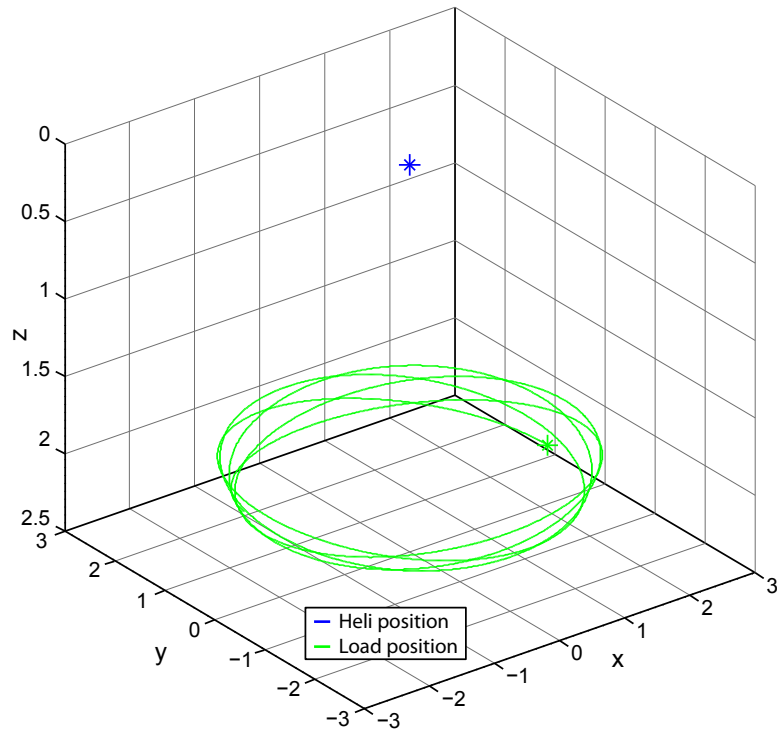


Figure 8.18: 3D plot showing the position of the helicopter and the slung load.

load start moving upwards, as well as around the inertial z-axis, this can be explained from the centrifugal force point outwards outweighing the gravity point downwards, resulting in the load moving up and down while also rotating around inertial z-axis. Had the centrifugal force been equal in size to the gravity, the load would not move up and down, but rotate in a perfect circle.

8.5.8 Conclusion of Rigid Body Model Simulations

Based on the simulations done in this section, the model is deemed to perform as expected throughout all 7 simulations. It is apparent that the model have been simplified when the attachment point of the slung load wire is in the helicopters centre of mass, since the helicopters heading this way is uncorrelated to the helicopters translatory movement, and the slung load entirely.

With the rigid body model performing as shown in this section through simulations, it is considered good enough for further use in the control part of this project, even though a verification against actual measurements are not available for this part of the model. The matlab files required to perform the simulations can be found on the DVD:



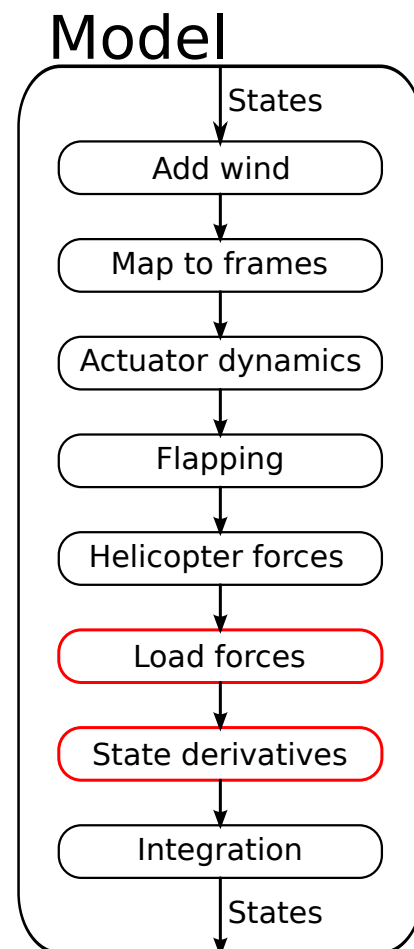
`/Model/Rigid body model/rbm.m`

`/Model/Rigid body model/calc_forces`

Model Implementation for Helisim

As mentioned in section 5.3 on page 27 in the helicopter model chapter about helisim, the semi-submerged load dynamics model and the rigid body model, have to be integrated into the source code of Helisim's model, which is written in C++. To achieve this two C++ functions are written, implementing each model. The function calls to Helisim's rigid body model and load dynamics models are then replaced in the source code.

The following is a brief explanation of how to interpret the sequence diagram in the figure to the right, and thereby give an overview of the procedure of the model implementation in Helisim. First a variable is created to hold the velocity of the helicopter, to avoid overwriting the actual velocity of the helicopter. Then the wind is mapped to the respective frames and added, resulting in a effective velocity that can be used to calculate lift and drag forces. Next, this velocity is mapped to the helicopter fixed frame and the tail frame. The acceleration is also mapped to helicopter frame, and the advance ratios of the main and tail rotor is calculated. The actuator dynamics are calculated, including motor dynamics. The flapping of the stabiliser bar and main rotor is calculated and the forces and torques generated by main and tail rotor. The drag on the helicopter is calculated and the helicopter forces are added up. The load forces and torques are calculated, and the state derivatives are determined. These state derivatives are then integrated up to gain the updated states. The functions for calculating the load forces and state derivatives are replaced by functions implementing the dynamics attained in this work. This is indicated in the figure to the right, by the red blocks. These functions are described in details in the following two sections.



9.1 Semi-submerged Load Dynamics Implementation

The source file for the implementation of the load dynamics can be found on the enclosed DVD, at the following path:



/Source/BuoyLoadFunctions.cpp

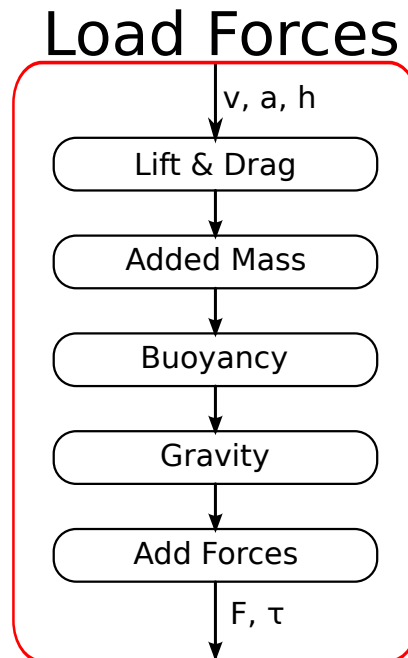
The file contains several functions that are used for the calculations. The model interface function prototype is shown below:

```
void SubSumOfForces(double *F_tau, double *v, double *a, double h);
```

The figure below shows the a sequence diagram of the flow in SubSumOfForces, and the following will describe the function more in detail.

The function takes four arguments. The first one `double *F_tau` is the return variable. It is a 6 element array containing the the resulting force and torque acting on the load in \mathbb{E} , where the torque is zero. `double *v` is a three element array containing the velocity of the load in \mathbb{E} , which is used for calculating the lift and drag force. `double *a` is a three element array containing the acceleration of the load in \mathbb{E} , which is used for calculating the added mass force. Lastly `double h` is a scalar value indicating the altitude of the load in, which is used in the calculations for the buoyancy, added mass, lift and drag forces.

The function call it replaces, returns the force and torque in the \mathbb{L} frame, but since it is only used by the rigid body model, witch in the case of this work requires the force in the \mathbb{E} frame, it is kept in the \mathbb{E} frame.



9.2 Rigid Body Model Implementation

The source file for the implementation of the rigid body model can be found on the enclosed DVD, at the following path:



/Source/BuoyRigidBodyModel.cpp

The file contains several functions that are used for the calculations. The model interface function prototype is shown below:

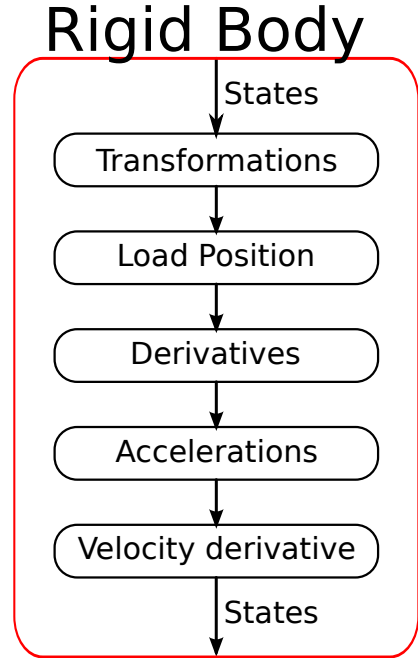
```
void BuoyLoadRBMInterface(double *Xo, double *Xi, double *Q, double *Acc);
```

The figure below shows the sequence diagram of the flow in the BuoyLoadRBMInterface function, and the following will describe the function more in detail.

The function takes four arguments. The first one `double *Xo` is a 44 element return variable, that holds the state time derivatives of entire system. The second one `double *Xi` is a 44 element input variable, that holds the states of entire system. `double *Q`, is a 12 element input variable that holds the forces and torques acting on the helicopter and load individually. `double *Acc` is a 6 element return variable that holds the accelerations of the helicopter and load.

Of the 44 states, only the first 24 are written in this function. These 24 elements are clustered in blocks of three, containing $\mathbb{E}\mathbf{p}_h$, $\mathbb{E}\boldsymbol{\theta}_h$, $\mathbb{E}\mathbf{p}_l$, $\mathbb{E}\boldsymbol{\theta}_l$, $\mathbb{H}\mathbf{v}_h$, $\mathbb{H}\boldsymbol{\omega}_h$, $\mathbb{L}\mathbf{v}_l$ and $\mathbb{L}\boldsymbol{\omega}_l$ in that order. Note that Helisim needs the position and velocity of the load, expressed in Cartesian coordinates and Euler angles and angular velocities, same as for the helicopter. These are not in the rigid body model presented in this work, but are calculated, based on position and velocity of the helicopter and the angles and angular velocity of the load. The angles and angular velocities of the load in Helisim are defined around the centre of mass of the load, but are only used for illustrative purposes by the graphics, and for calculation of the load forces. Since the function calculating the load forces have been replaced by the one described in the above section, the only implication of moving the point of which the load rotates around to the attachment point on the helicopter, is that the load rotates around its centre

of mass in the graphics. One other implication of calculating the load position inside this function is that it is updated based on the load angles, instead of by integration of the velocity, and therefore is updated one sample later than the rest of the states.



Summary of part II

The first chapter of the model part introduced the helicopter model that was to be used with the modelling of this work, and further in the control part of this work.

The slung load modelling included a series of the forces applied on the slung load, including the depth dependency of most of the applied forces.

The rigid body model was derived to obtain the expression of both translational and angular accelerations. This was done by applying Euler's laws of motion on the slung load and helicopter, and using the expression of acceleration in the helicopter frame. Some simplifications to the model were made in the rigid body modelling chapter, including assumptions of the load wire being tense at all time, and having the slung load as a point mass.

In the model simulation chapter all the forces on the slung load were implemented in a MATLAB simulation, compared to a set of experiment measurements, to verify the model. The rigid body model simulations showed how the slung load and helicopter affect each other in different scenarios, and what the modelling simplifications meant for the results. The chapter of the part explains how the modelling done in this work was implemented to work with the Helisim project.

Part III

Controller

Part introduction

Contents

10 General Control Strategies

11 Linear Quadratic Control

12 Model Predictive Control

Purpose

The purpose of this part is to explain and use a series of control strategies, both on a standalone helicopter and on the complete helicopter with slung load system.

The first chapter of this part includes general control strategies, specifying the different control strategies used on the system, including operating points found through trimming, and ranges for the control states and input signals, as well as a linear model analysis of the uncontrolled system to be used in the LQ chapter.

The second chapter of this part is the methodology and use of Linear Quadratic Control, first deriving controllers for a standalone helicopter in hover and in forward flight, with a heading reference, and afterwards designing LQ controllers with the same capabilities, but for the complete system.

The last chapter deals with Model Predictive Control, which is implemented in cascade with a LQ controller. The inner loop LQ control is designed to stabilise the system, which is checked using pole analysis of the closed inner loop, and the outer MPC loop is set to control the set-points for the LQ controller.

Goals

- *Explain the methodology of the different control strategies used on the system.*
- *Create a series of controllers that is able to control a standalone helicopter, to a degree where it is able to adhere to the confines of the requirement specification.*
- *Create a series of controllers that is able to control a helicopter with the submerged slung load attached, to a degree where it is able to adhere to the confines of the requirement specification.*

General Control Strategies

This chapter presents the different control strategies that are examined in this work, along with the necessary considerations and decisions each of the control strategies require for a controller to be produced. These considerations include ranges for each of the states and operating points for linearisation.

The first of the control strategies used in this work is Linear Quadratic (LQ) control, which is a single feedback matrix found from a linearised model. LQ controllers are found for the helicopter in hover and in forward flight with a yaw angle reference, both for a standalone helicopter, and a helicopter with slung load system. Before the control is done, the linearised system poles are analysed, to find out if the system has unstable dynamics. The LQ controllers should be implemented as shown in figure 10.1, with the necessary information to do so presented in section 10.1. LQ control is chosen for its ability to handle MIMO systems well, as well as for its simplicity in the design process.

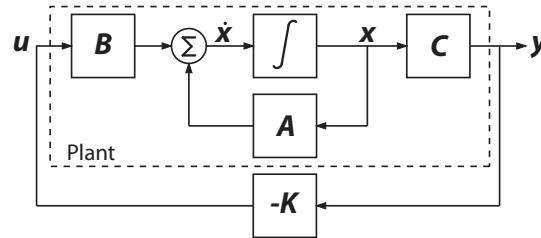


Figure 10.1: Diagram of the state space representation of the linearised system, with an LQ controller as feedback matrix.

The second control strategy is model predictive control (MPC) implemented as a set-point controller to an underlying LQ controller. First the dynamics of the closed loop LQ system are examined, to find out whether the MPC is implemented on a stabilised system, since that has a large impact on how fast the model predictive controller has to be. The MPC is to be implemented as shown in figure 10.2. MPC is chosen as a control strategy for this work, both for its ability to handle MIMO systems with constraints and disturbances, but also for its ability to predict the behavior of the system over a time horizon, making it able to handle transitions in set-point changes smoothly.

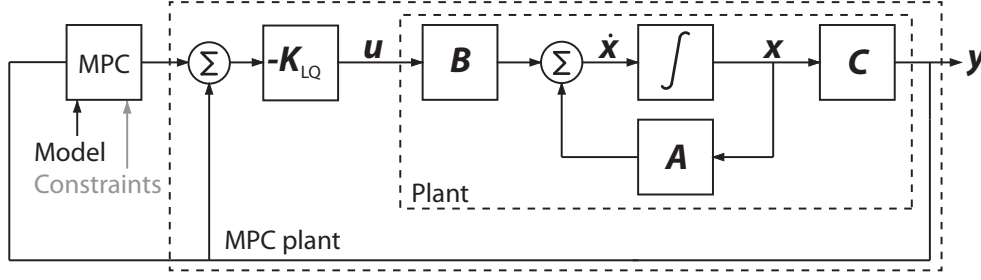


Figure 10.2: Diagram of the state space representation of the linearised system, with an LQ controller as feedback matrix and MPC to control the LQ references.

10.1 LQ Control Considerations

Since LQ control requires a linearised model to be produced, the first step in this particular control strategy is to determine the operating points where the produced controller should work, and the ranges of the deviations from the operating point for each state, to be used in the initial weight matrices for the controller. This is to be done for both a helicopter in hover and in flight. A summary of how an LQ controller is produced is included in section 11.1 on page 85.

The operating point and range considerations for the LQ controller are split up into a standalone helicopter, and a slung load subsection, to give a better overview of which states belong to what part of the system.

10.1.1 Helicopter Operating Points and Ranges

First, all the states need operating points where the linearisation should take place. In the reduced order model presented by [7], the system has 12 states, and 4 inputs. If main rotor flapping, stabilizer bar flapping, actuator dynamics and motor dynamics all are modelled as steady-state. The trimming and linearisation of the system are presented in appendix B on page 179.

The trimmed values for the position and angles of the standalone helicopter, as well as the values for the control signals are shown in table 10.1 and 10.2. The helicopter is in both cases trimmed to fly at an altitude of 6 m, and the flight controller is trimmed for a x-velocity of 6 m/s. The trimmed state and control signal values are used together with the LQ controller as shown in figure 10.3. The remaining velocity states are all trimmed to be 0.

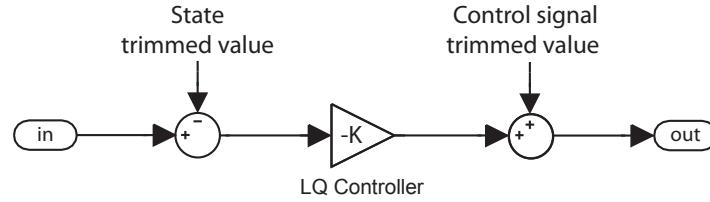
The operating points presented in this section are used to linearise the reduced order helicopter model in appendix B.

Apart from the linearised model and operating points, a set of operation ranges are needed, in which the controller should keep the state values. These values should all be specified as deviations from the operating points, and is not the operational range

Trimmed state values						
	x_h	y_h	z_h	ϕ_h	θ_h	ψ_h
Hover	0.0000	0.0000	-6.0000	0.0517	-0.0011	0.0000
Flight	0.0000	0.0000	-6.0000	0.0415	-0.0035	0.0000

Table 10.1: Trimmed state operating points.

Trimmed control signals				
	θ_{col}	θ_{lat}	θ_{lon}	ω_{tr}
Hover	0.1946	-0.0032	-0.0013	0.1298
Flight	0.1777	-0.0027	0.0063	0.0901

Table 10.2: Trimmed control signals.**Figure 10.3:** Use of the trimmed state and control signals with the LQ controller.

of the states, but rather the allowed deviation from the operating points by the LQ controller. These operation point deviations is to be used for the initial weightings on each state, in the LQ controller design. Control signals for both hover and flight controllers on a standalone helicopter is shown in table 10.3, which goes together with table 10.4, showing the expected ranges of the helicopter states.

	Hover range	Flight range	Unit
θ_{col}	$\pm \pi/8$	$\pm \pi/8$	rad
θ_{lat}	$\pm \pi/16$	$\pm \pi/16$	rad
θ_{lon}	$\pm \pi/16$	$\pm \pi/16$	rad
ω_{tr}	$\pm \pi/4$	$\pm \pi/2$	rad/s

Table 10.3: Control signal operating point deviation ranges for both the hover and flight controller.

The first 3 control signal ranges are all controls for the helicopters main rotor, specifically the collective-, lateral- and longitudinal blade pitch, all specified in radians, with the last control signal being a reference of the helicopter yaw rate, controlled by the tail rotor, given in rad/s.

	Hover range	Flight range	Unit
x_h	± 0.5	-	m
y_h	± 0.5	-	m
z_h	± 0.2	± 0.2	m
ϕ_h	$\pm \pi/4$	$\pm \pi/4$	rad
θ_h	$\pm \pi/4$	$\pm \pi/4$	rad
ψ_h	$\pm \pi/2$	$\pm \pi/8$	rad
$v_{x,h}$	± 2	± 0.2	m/s
$v_{y,h}$	± 2	± 0.4	m/s
$v_{z,h}$	± 0.5	± 0.5	m/s
$\omega_{\phi,h}$	$\pm \pi$	$\pm \pi$	rad/s
$\omega_{\theta,h}$	$\pm \pi$	$\pm \pi$	rad/s
$\omega_{\psi,h}$	$\pm \pi$	$\pm \pi$	rad/s

Table 10.4: State operating point deviation ranges for both the hover and flight controller.

The values in table 10.4 is initial thoughts on how large deviations from the operating points that should be allowed by the LQ controllers. The dash in x- and y-position of the flight ranges indicate that deviations in those states should not matter to the controller. Since the ranges, and through those the state weighting, is not hard constraints some of the ranges is set to allow less deviation than what is specified in the requirement specification.

10.1.2 Helicopter with Slung Load Operating Points and Ranges

When a load is attached to the helicopter, it changes the dynamics of the system. These changes has to be accounted for in the trimmed values, and thus new values are needed. The hover trim is done for an altitude of 3 m, which is also the length of the wire, leaving the slung load in the water surface, just above its equilibrium. In this case where the slung load is being lifted above its steady state position, the modelling simplifications assuming a rigid wire can be justified. The flight trim is done for a forward velocity of 2 m/s. The trimmed values are shown in tables 10.5 and 10.6.

Trimmed state values						
	x_h	y_h	z_h	ϕ_h	θ_h	ψ_h
Hover	0.0000	0.0000	-3.0000	0.0500	-0.0000	0.0000
Flight	0.0000	0.0000	-3.0000	0.0500	-0.0001	0.0000

Table 10.5: Trimmed state operating points.

Trimmed control signals				
	θ_{col}	θ_{lat}	θ_{lon}	ω_{tr}
Hover	0.2052	-0.0007	-0.0013	0.1001
Flight	0.2051	-0.0009	0.0007	0.1000

Table 10.6: Trimmed control signals.

With the load attached, 4 additional system states are introduced in the system, being the Euler angles describing the position of the load, and angular velocities. The 4 additional states should have operation point deviations just like the rest of the states, these are shown in table 10.7.

	Hover range	Flight range	Unit
θ_l	$\pm \pi/4$	-	rad
ψ_l	$\pm \pi/4$	-	rad
$\omega_{\theta,l}$	$\pm \pi/2$	$\pm \pi$	rad/s
$\omega_{\psi,l}$	$\pm \pi/2$	$\pm \pi$	rad/s

Table 10.7: State operating point deviation ranges for both the hover and flight controller.

The range of the loads angles while in flight should just like the helicopter position not matter to the controller, since the load is expected to be dragged behind the helicopter at flight, while it should be minimised in hover. The operating points, state- and control signal ranges presented in this and the previous section will be used to produce a series of LQ controllers in chapter 11 on page 85.

10.1.3 Standalone Helicopter Linear Model Analysis

With the helicopter model linearised in appendix B, it is relevant for further use of the linearised model to analyse the poles of the system, to understand the dynamics of the uncontrolled linearised system. The first model presented in appendix B is trimmed for hover prior to the linearisation, meaning the poles presented here, along with their corresponding natural frequency and damping ration are for the standalone helicopter in hover, all the pole information is shown in table 10.8.

The linearised model for the helicopter in hover contain a pole pair with positive real values, meaning the system is unstable without control, as well as a pole pair very close to the zero real value axis, in the region where that pole pair should be controlled as well. The pole pairs seems to come from the pitch and roll angular velocity parts of the system, which means the helicopter most likely would crash without control on these states.

Description	Pole	ζ	ω_n
Fourth order pole	0	-	0
Single pole	-45.85	1	45.85
Single pole	-7.24	1	7.24
Single pole	-4.31	1	4.31
Pole pair	$0.22 \pm 0.62i$	-0.33	0.66
Pole pair	$-0.21 \pm 0.68i$	0.29	0.71
Single pole	-0.97	1	0.97

Table 10.8: List of poles and their corresponding natural frequency and damping ratio.

With the poles of the system found for hover, revealing a series of them having either positive real values, or very close to, it is also relevant to look at the pole movement, when hover is changed to forward flight, at different velocities. The pole movement over a sweep from 0m/s to 10m/s forward velocity shown in figure 10.4.

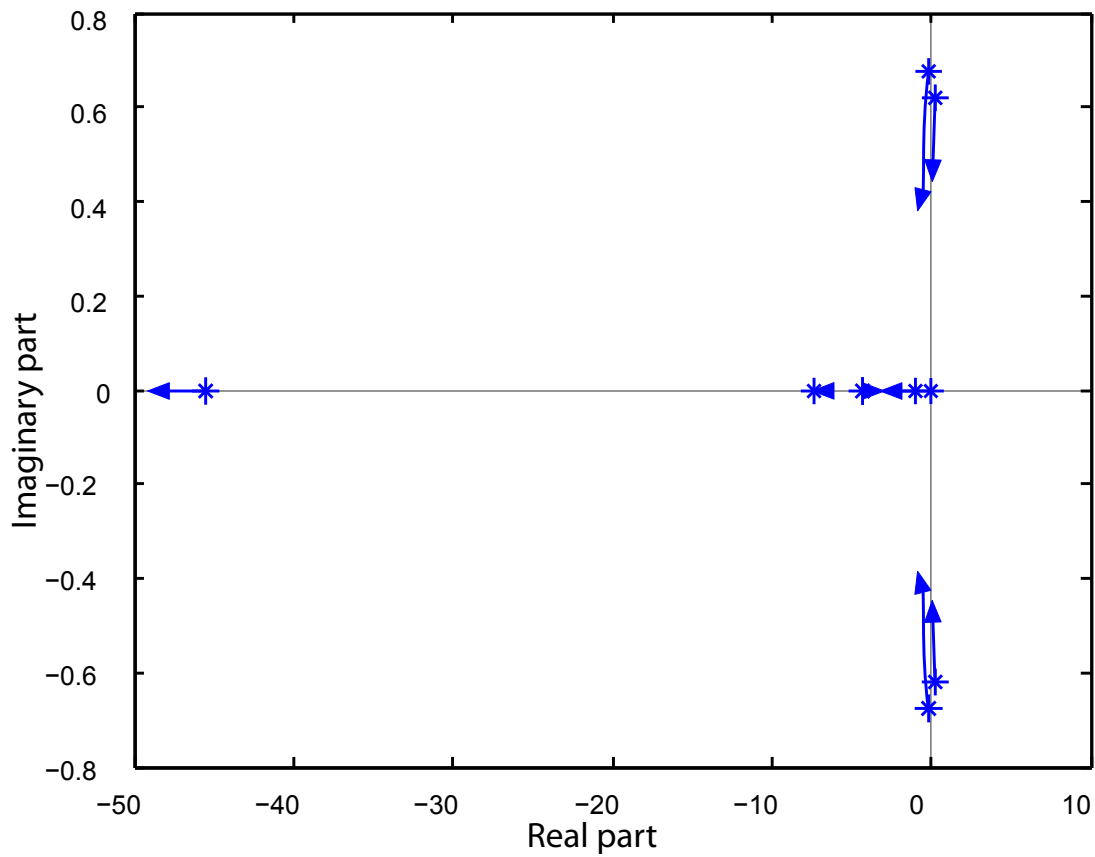


Figure 10.4: Pole plot of the standalone helicopter, with pole movement at the flight velocity varying from 0m/s through 10m/s .

The pole movement show that when the helicopter starts moving forward, the pitch and roll angular velocity poles does become slightly more stable, but with one pole pair still with positive real values, and the other being very close, even at 10m/s and still require control for the system to become stable. The linearised model is used to make the LQ controllers in chapter 11 on page 85.

10.1.4 Helicopter with Slung Load Linear Model Analysis

The system changes dynamics when a slung load is attached to the helicopter, especially in a case like the one in this work where the load is subjected to large forces when being dragged, which ultimately affect the helicopter as well. The procedure used to find a linear model for the helicopter with semi-submerged slung load, is presented in the second section of the linearisation appendix B on page 179. The linear model is used to observe the poles of the system in hover, and the pole movement when the helicopter changes from hover to flight. Table 10.9 show the poles of full system in hover.

Description	Pole	ζ	ω_n
Third order pole	0	-	0
Single pole	-47.18	1	47.18
Single pole	-6.65	1	6.65
Single pole	-3.86	1	3.86
Single pole	-4.52	1	4.52
Single pole	3.09	1	3.09
Pole pair	-0.08±0.79i	0.10	0.80
Pole pair	0.06±0.76i	-0.08	0.77
Pole pair	-0.09±0.59i	0.16	0.59
Pole pair	0.12±0.56i	-0.21	0.57

Table 10.9: List of poles and their corresponding natural frequency and damping ratio.

The table show that the system with slung load attached has become more unstable when compared to the standalone helicopter pole analysis, with 2 unstable pole pairs, 2 pole pair that is barely stable, and a single unstable pole at 3.09, meaning that the system require control to stabilise.

A pole movement plot have been produced for a sweep of forward flight velocity from 0m/s through 10m/s, to find out if the system stabilises when the system goes from hover to forward flight. The pole movement is shown in figure 10.5.

The pole movement plot show how the helicopter with slung load system partly destabilises while flying forward, showing that the system require control on multiple states with the slung load attached. It should be noted that the poles presented here are not those of the real system, but of the Bisgaard helicopter model [7] expanded with slung load and rigid body modelling from this work, including the model simplifications made in the process.

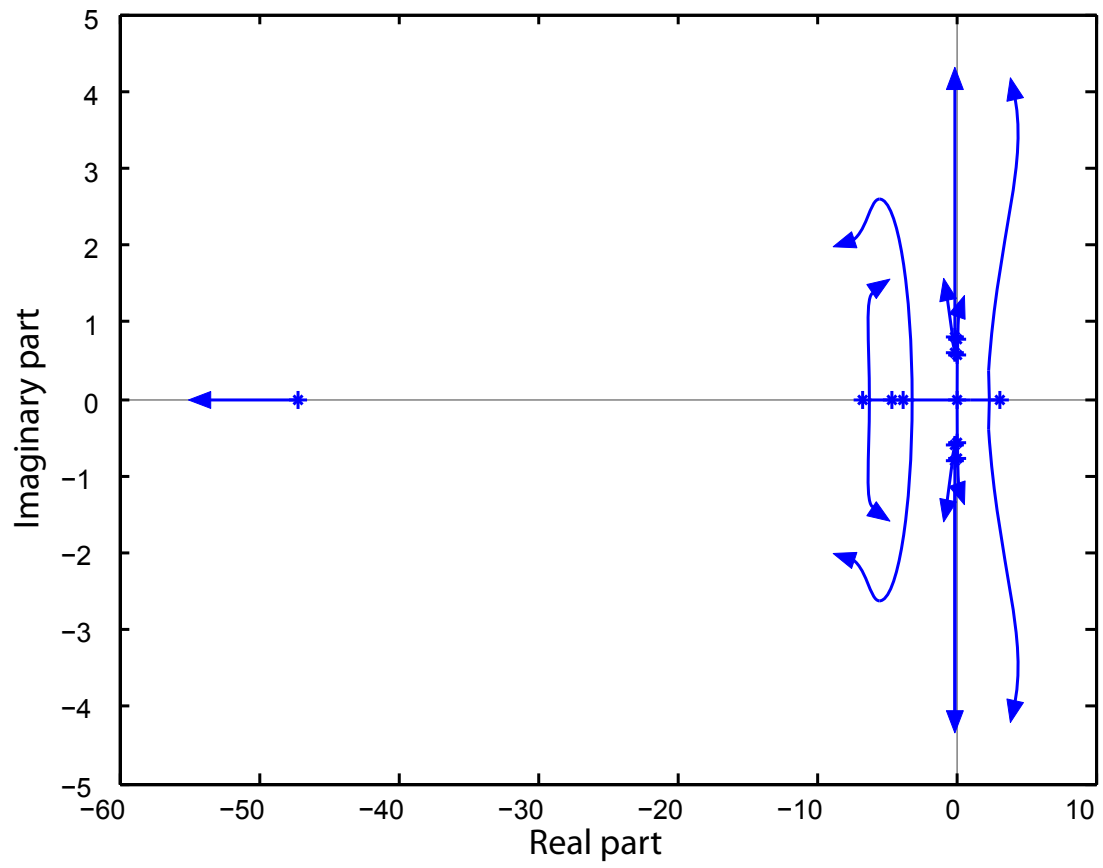


Figure 10.5: Pole plot of the helicopter with slung load, with pole movement at the flight velocity varying from 0^{m/s} through 10^{m/s}.

10.2 Model Predictive Control Considerations

The model predictive controller does, as explained in the beginning of the chapter, not control the system directly, but rather the set-points for an underlying LQ controller designed to stabilise the system. The system seen by the MPC is equal to the closed loop system seen by the LQ controller. This type of cascade control is according to [16, p.27] a very normal application of model predictive control, allowing it to have the longer computational time that is often associated with model predictive control. It is however not a requirement to have underlying controllers, but it is an advantage to implement model predictive control on a stable system if long prediction and control horizons are wanted, this is discussed in the MPC chapter 12 on page 113.

When working with cascade control it is important to allow the inner loop controller time to react to set-point changes from the outer control loop. Because of this a rule of thumb is that the inner loop should run 4-10 times faster than the outer control loop, with sufficient bandwidth to follow the reference from the MPC. In this case the sample length of the inner loop is known to be 50 Hz, which is the sample frequency of the estimated sensor values in Helisim, so a sample frequency for the MPC is chosen to be 5 Hz, to give the controller time for computations, leaving the sample length $MPCTs$ at 0.2 s.

The prediction horizon H_p and control horizon H_u , is respectively the amount of time steps for which the system responses based on a linearised model should be generated, and the amount of time steps control signals should be calculated for. These two horizons is a part of the MPC considerations, as well as part of the controller design itself, since longer horizons can produce better results for the controller, but increases the computational time the controller needs at each time step, which is of course based on the available computational power for the system. Since the specific number of samples used for the prediction and control horizons is not final before the tuning of each controller is done, a finite number can not be given here. It can however be speculated that longer horizons is needed for the flight controller if velocity and yaw references are changed, than for the hover controller stabilising the system at a specific position.

The considerations made here is to be used in the production of a series of model predictive controllers in chapter 12, where the general idea and methodology behind the model predictive control strategy is explained as well.

Linear Quadratic Control

This chapter introduces the first of the control strategies used to control the helicopter with slung load system, namely Linear Quadratic (LQ) control. However, an LQ controller requires a state space model, linearised around a set of operating points, for the controller to be applied. The linearisation is handled in appendix B on page 179 using the operating points presented in the control strategies chapter 10 on page 75.

The methodology used to compute the LQ controller, along with the considerations and requirements that must be fulfilled to use LQ control are presented in section 11.1 below. The computed controller is then evaluated in the acceptance test chapter 13 starting on page 141, where it is used on the helicopter with slung load model in Helisim.

11.1 LQ Control Methodology

Before any controller can be found, the system which the controller is to be applied on, needs to be examined for controllability. Controllability in other words is verifying whether the \mathbf{A} and \mathbf{B} is a controllable pair, by checking if all the states in the model can be controlled through the matrix pair. The standard way to check for controllability is by computing the controllability matrix \mathcal{C} shown in (11.1), and then check if the controllability criterion is fulfilled as shown in (11.2).

$$\mathcal{C} = [\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A}^2\mathbf{B} \quad \dots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (11.1)$$

$$\text{rank}(\mathcal{C}) = n \quad (11.2)$$

\mathcal{C}	Controllability matrix.
\mathbf{A}	System state matrix.
\mathbf{B}	Input matrix.
n	Number of states in the model.

If the controllability matrix is shown to have full rank, the controllability criterion is achieved, and an LQ controller can be designed for the system.

11.1.1 Design of a Linear Quadratic Controller

LQ control is a subsection of optimal control theory that work to find the controller that optimise the performance of a system in the operating points. This is done by finding a controller that minimise a performance function containing a pair of cost matrices expressing costs associated with changes from the operating point for each of the states in the system.

The performance function presented in (11.3) is the function for an infinite-horizon continuous time system. Infinite horizon is needed due to the unknown duration of a SAR mission.

$$\mathcal{I} = \int_0^{\infty} x(t)^T \mathbf{Q} x(t) + u(t)^T \mathbf{R} u(t) dt \quad (11.3)$$

\mathcal{I}	Continuous performance function.
$x(t)$	State values at time t .
$u(t)$	Control signals at time t .
\mathbf{Q}, \mathbf{R}	Weight matrices for the states and inputs.

To find the controller that minimise the performance function, the weight matrices \mathbf{Q} and \mathbf{R} first have to be found. These weight matrices are both square matrices with dimensions $\mathbf{Q}_{n \times n}$ and $\mathbf{R}_{m \times m}$ with n being the number of states and m the number of input signals to the system with \mathbf{Q} being positive semi definite, and \mathbf{R} positive definite. The positive definiteness is required in order to ensure a positive definite solution to (11.7). If the weight matrices are designed correctly, it is possible to have strict constraints on some states, while allowing large derivations in others, or having some control signals changing more than others.

While it is possible to weight multiple states or control signals together when designing \mathbf{Q} and \mathbf{R} , the most common method of designing the weight matrices are to only use the diagonal entries, resulting in weights on every state and control signal individually. One way to calculate the diagonal entries in the initial weight matrices is by Bryson's rule, which is presented in (11.4).

$$\mathbf{Q}(i, i) = \frac{1}{x_{i,max}^2} \quad \mathbf{R}(j, j) = \frac{1}{u_{j,max}^2} \quad (11.4)$$

$x_{i,max}$	Largest value a state x_i is allowed to become.
$u_{j,max}$	Largest value a control signal u_j is allowed to become.

The method shown in (11.4) is presented in [1], and it should be noted that the resulting weights only are preliminary values, which often need a series of iterations to yield the

wanted results. As the \mathbf{Q} and \mathbf{R} matrices has been found using the method from (11.4), and as the control law is assumed as (11.5), it is now possible to find a feedback matrix \mathbf{K} for the LQ controller, that minimise the performance function from (11.3).

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \quad (11.5)$$

The \mathbf{K} matrix is found by solving (11.6), with \mathbf{Q} , \mathbf{R} , \mathbf{A} and \mathbf{B} already known, and \mathbf{S} being the solution to the continuous time algebraic Riccati equation (CARE) in (11.7).

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} \quad (11.6)$$

$$\mathbf{0} = \mathbf{Q} + \mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{S} \quad (11.7)$$

K	Feedback matrix to control the system.
S	Solution to the continuous time algebraic Riccati equation.
A	State matrix.
B	Input matrix.

Then the \mathbf{A} and \mathbf{B} matrices are known, along with their corresponding weight matrices \mathbf{Q} and \mathbf{R} , the feedback matrix \mathbf{K} can be found in MATLAB using the `lqr(A,B,Q,R)`-function that minimises the performance function. This is also appended on the DVD:



/Controller/SAH/SAH_hover.mat
/Controller/SAH/SAH_flight.mat

11.2 LQ Control of a Standalone Helicopter

The first controller that is computed is for a standalone helicopter, that should be controlled so it is able to hover around a specific position. This controller is being computed to fulfil the standalone helicopter system requirements from section 4.2.

The controllability matrix \mathcal{C} is found to have the rank 12, which means the system is controllable, and the design process of the LQ controller can proceed to the initial \mathbf{Q} and \mathbf{R} weight matrices.

11.2.1 Finding the Weight Matrices

With the methodology explained, and ranges for the states and input signals determined in subsection 10.1.1 on page 76, for a scenario with the helicopter without a slung load, being in hover.

When only finding diagonal entries of the weight matrices, to keep it clear which weights should be changed if the computed controller does not behave as expected. Using the linearised helicopter model from appendix B, with 12 states and 4 inputs, the \mathbf{Q} and

\mathbf{R} matrices will be on the form seen in (11.8), with the diagonal entries being found subsequently.

$$\mathbf{Q} = \begin{bmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & q_{12} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_4 \end{bmatrix} \quad (11.8)$$

The \mathbf{Q} matrix have a diagonal entry for all of the 12 states presented in subsection 10.1.1 in table 10.4 on page 78, and the initial estimate for the entries are found using the method from (11.4).

$$\mathbf{Q}_{diagonal} = [4 \ 4 \ 25 \ 1.62 \ 1.62 \ 0.41 \ 0.25 \ 0.25 \ 4 \ 0.10 \ 0.10 \ 0.10] \quad (11.9)$$

With all the diagonal entries for \mathbf{Q} shown in (11.9), a initial estimate for \mathbf{Q} have been generated from table 10.3 in the general control strategies chapter 10 on page 75. The \mathbf{R} weight matrix have 4 entries, one for each of the inputs.

$$\mathbf{R}_{diagonal} = [6.48 \ 25.94 \ 25.94 \ 1.62] \quad (11.10)$$

With the \mathbf{Q} entries found earlier, and \mathbf{R} weights shown in (11.10), and the state space representation of the model presented through the \mathbf{A} and \mathbf{B} matrices in appendix B, all the information needed to compute a controller has been found.

11.2.2 Deriving a Controller for a Helicopter in Hover

In this controller of the standalone helicopter, it is expected that all states can be measured, so that full state feedback is possible without an observer. In this case that means that the state values are the same as the system outputs, and the normal state space representation block diagram with feedback control can be cut down to the block diagram in figure 11.1.

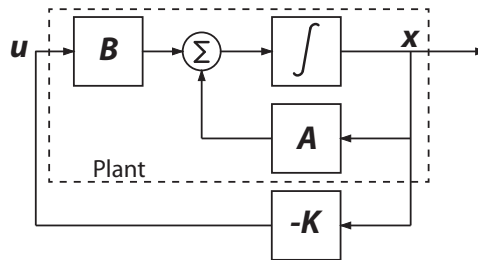


Figure 11.1: A block diagram of the model in state space representation with a feedback control matrix \mathbf{K} implemented.

Using the \mathbf{A} and \mathbf{B} matrices from appendix B, together with the \mathbf{Q} and \mathbf{R} weight matrices found in (11.9) and (11.10) it is now possible to find a feedback matrix \mathbf{K} that optimises the performance function with the given weightings, named $\mathbf{K}_{hover,initial}[4 \times 12]$.

The controller is implemented into a stripped version of the Simulink Helisim model, as shown in figure 11.2.

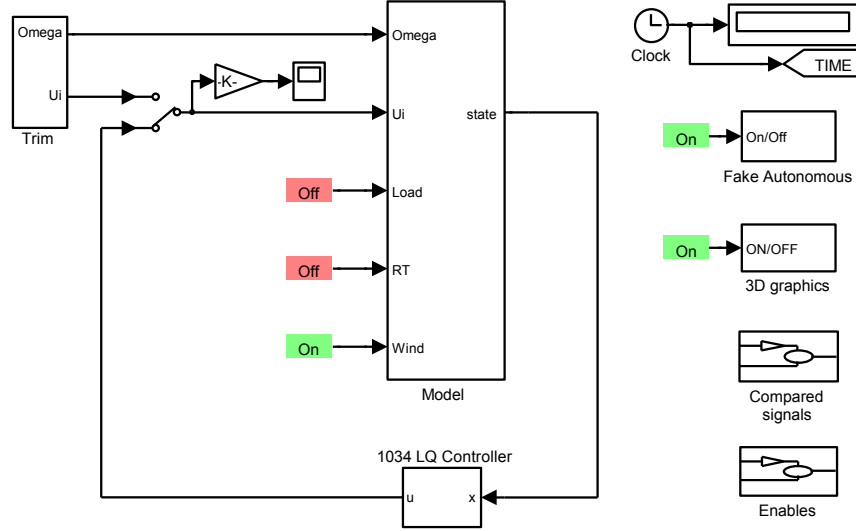


Figure 11.2: An illustration of the stripped version of Helisim, with the feedback controller implemented.

Since the feedback matrix $\mathbf{K}_{hover,initial}$ control on x- and y-position on the helicopter, but is computed from a model, linearised with a yaw angle of 0, a problem rises when the yaw angle goes far away from that operating point. With the x- and y- position of the helicopter defined in the earth fixed frame, and the x- and y- velocities in the helicopter fixed frame, the problem becomes apparent then the helicopter is turned to a yaw angle of π rad, now the relations between the position and velocity has opposite signs in a non-linear model, compared to a yaw angle of 0 rad, but the linearised model does not take care of that.

A solution to that problem is to rotate the x- and y-positions for the feedback matrix with the yaw angle of the helicopter, in order to trick the controller to give the correct responses through the control signals. The x-, y-position rotation is done as shown in (11.11).

$$\begin{bmatrix} x_{rot} \\ y_{rot} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (11.11)$$

With the controller implemented into Helisim, simulations revealed that the controller is able to stabilise the helicopter in hover to some extent, but with control signals oscillating to a degree where it if tested on a real helicopter would break down the mechanics or crash, even while simulating without wind.

To get better results with the controller, some tuning of the weights is required. The problem has been found to be caused by heavy restrictions in the \mathbf{Q} matrix on some of the velocities and angular velocities of the hovering helicopter. Better results are achieved if the lateral and longitudinal blade pitch is weighted heavier than the collective blade pitch in the \mathbf{R} matrix. The new weightings for \mathbf{Q} and \mathbf{R} are shown in (11.12).

$$\begin{aligned}\mathbf{Q}_{diagonal} &= [1 \ 1 \ 20 \ 1.62 \ 1.62 \ 20 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01] \\ \mathbf{R}_{diagonal} &= [50 \ 100 \ 100 \ 50]\end{aligned}\quad (11.12)$$

The new feedback control matrix \mathbf{K} obtained much better results than the initial control matrix. The new iterations of \mathbf{K}_{hover} , achieved the position results shown in 11.3 with the control signals shown in 11.4 and a wind disturbance as shown in figure 11.5. The position results are shown as deviations from the operating points.

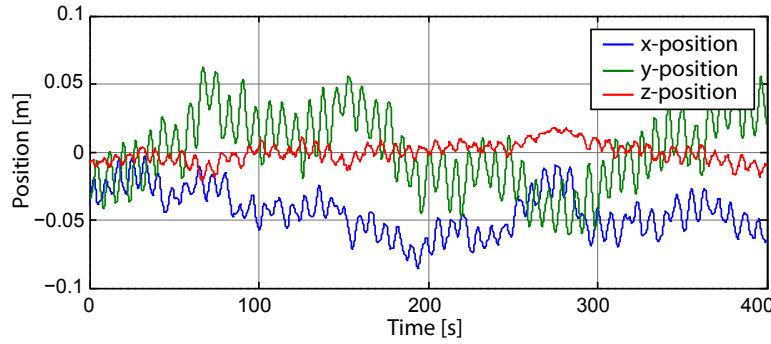


Figure 11.3: Plots of the helicopters position, plotted for a period of 400 seconds. The hover altitude has been subtracted from the z-position.

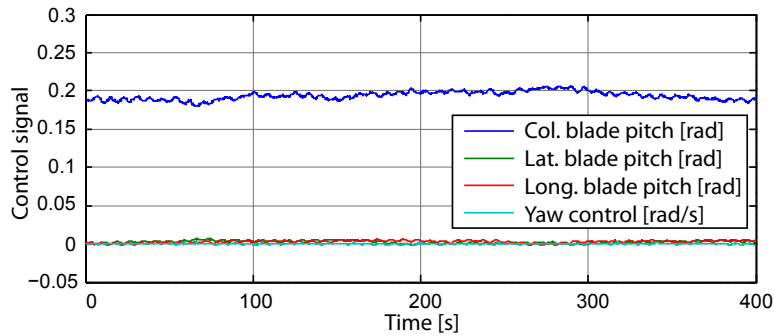


Figure 11.4: Plots of the 4 control signals for a period of 400 seconds.

The performance plots show that the main disturbance of the system with perfect state feedback is the wind, where the changes in position match the changes in wind closely. The controller weights presented here has already been iterated for performance on a system with measurement inaccuracy and noise, and is thus the same controller used in the LQ part of the acceptance test. The final \mathbf{K} is presented in (11.13) and in

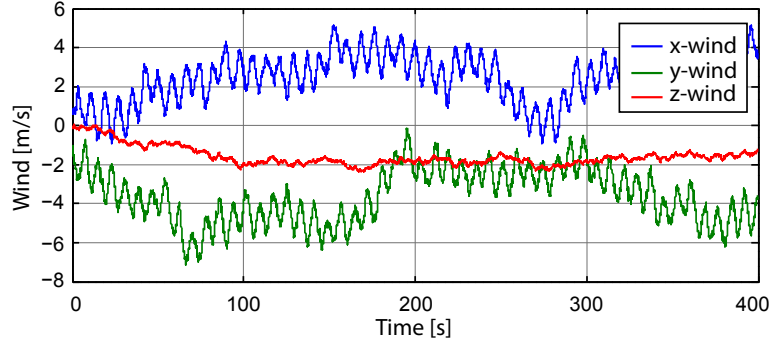


Figure 11.5: Plots of the 3 wind components for a period of 400 seconds.

SAH.hover.mat on the DVD. Note that the \mathbf{K} matrix is shown transposed.

$$\mathbf{K}_{hover,final}^T = \begin{bmatrix} 26.87 \cdot 10^{-3} & -48.84 & 87.26 & 566.17 \cdot 10^{-3} \\ -397.96 \cdot 10^{-3} & -86.37 & -48.28 & -20.45 \\ -631.53 & 4.04 & 2.47 & -33.54 \\ -407.71 \cdot 10^{-3} & -206.21 & -125.46 & 4.34 \\ -2.41 & 153.25 & -316.10 & -33.21 \\ -34.16 & -55.42 & -33.04 & 624.91 \\ 226.33 \cdot 10^{-3} & -39.78 & 73.62 & 1.70 \\ -4.72 & -59.35 & -36.11 & 637.38 \cdot 10^{-3} \\ -84.71 & 5.09 & 1.94 & -26.52 \\ 20.15 \cdot 10^{-3} & -5.75 & -1.16 & 8.15 \cdot 10^{-3} \\ -188.67 \cdot 10^{-3} & 8.36 & -46.52 & 74.51 \cdot 10^{-3} \\ 929.63 \cdot 10^{-3} & -19.07 & -7.35 & 83.59 \end{bmatrix} \cdot 10^{-3} \quad (11.13)$$

The performance of this final hover controller without perfect state feedback can be seen in its use in the first acceptance test 13.1 on page 142.

11.2.3 Deriving a Controller for a Helicopter in Flight

While the design process and implementation of a flight controller should be near identical to the one used to find the controller for a helicopter in hover, some changes has to be made to the weightings in both the \mathbf{Q} and \mathbf{R} matrices, since the controller should be able to fly a standalone helicopter to a given position, with a constant velocity. The initial weightings shown in (11.14) for the flight controller are found from tables 10.4 and 10.3, just like with the initial weights for the hover controller.

$$\begin{aligned} \mathbf{Q}_{init,diagonal} &= [10^{-10} \ 10^{-10} \ 25 \ 1.62 \ 1.62 \ 6.48 \ 25 \ 6.25 \ 4 \ 0.10 \ 0.10 \ 0.10] \\ \mathbf{R}_{init,diagonal} &= [6.48 \ 25.94 \ 25.94 \ 0.41] \end{aligned} \quad (11.14)$$

The weightings for a controller of a standalone helicopter with a constant velocity in a specific direction has to have very low weightings on the x- and y-position, but a

stricter weighting on the x-velocity and yaw heading. The helicopter is unable to fly with the initial weightings, which had to be changed, specifically in the weightings of the control signals that had to be significantly higher. The weightings for the new flight controller are shown in (11.15).

$$\begin{aligned} \mathbf{Q}_{diagonal} &= [10^{-10} \ 10^{-10} \ 10 \ 1 \ 1 \ 3 \ 1 \ 1 \ 0.01 \ 0.01 \ 0.01 \ 0.01] \\ \mathbf{R}_{diagonal} &= [40 \ 800 \ 800 \ 5] \end{aligned} \quad (11.15)$$

To be able to fly towards a specific location, the reference position is given and the difference between that and the current position is converted into a yaw angle offset that is added to the operating point of the yaw angle. Since the offset to the yaw angle is wanted in the range $[-\pi, \pi]$ an arctangent variation called atan2 is used. atan2 is a function that takes two arguments, which here is the difference between the current position and the reference position in x- and y-coordinates including the sign of the distances. This implementation is shown in the top left corner in figure 11.6.

Use of the atan2 function does however rise a problem if the helicopter is directly in front of the reference position in the x-direction seen in the earth fixed frame. Since the function will alternate between $\pm\pi$. To account for this problem an if-function is introduced, checking whether the difference between the atan2 function and the current helicopter yaw angle is above π , if this is the case the helicopter is set to turn the other way, since that is going to be the shortest. However, this way the yaw angle of the helicopter can get above 2π , and a modulus 2π is added to the helicopter yaw angle, since it is wanted in the original range.

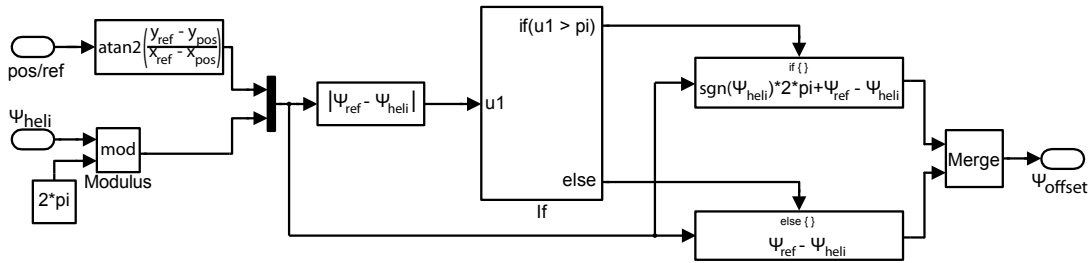


Figure 11.6: A Simulink implementation of the the yaw offset based on a reference position.

The performance of the flight controller for the yaw angle and the x-velocity given perfect state feedback is shown in figure 11.7 and 11.8 with the control signals shown in figure 11.9. The wind used in this simulation is equal to that of the hover simulation shown in figure 11.5.

The performance plots show as with the hover controller that the main disturbance in the system with this setup is the wind, here shown through the changes in yaw-angle and x-velocity, where the changes in those states follow the changes in wind closely. The weightings used in the controller has already been iterated to work on the system while having estimated sensor values instead of the perfect state feedback shown

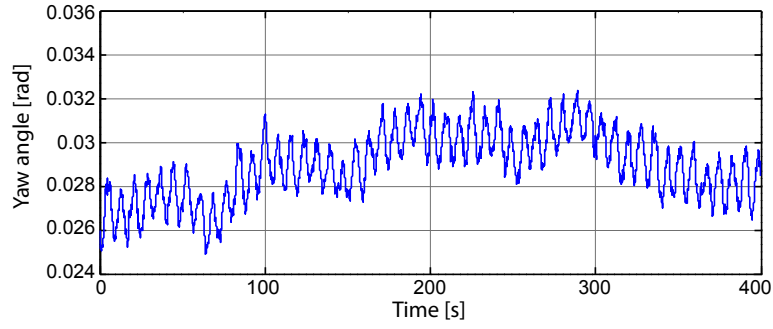


Figure 11.7: The yaw angle state for a period of 400 seconds.

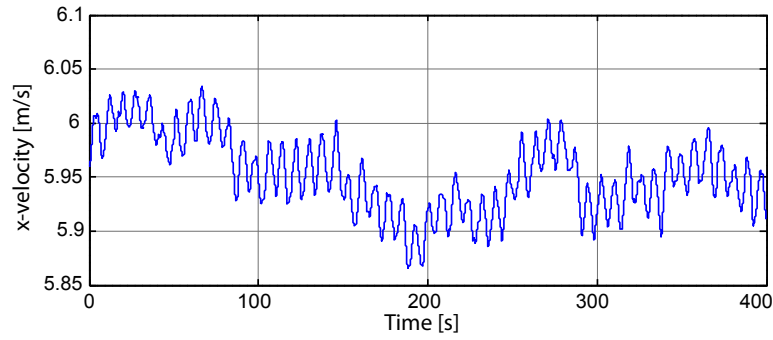


Figure 11.8: The x-velocity state for a period of 400 seconds.

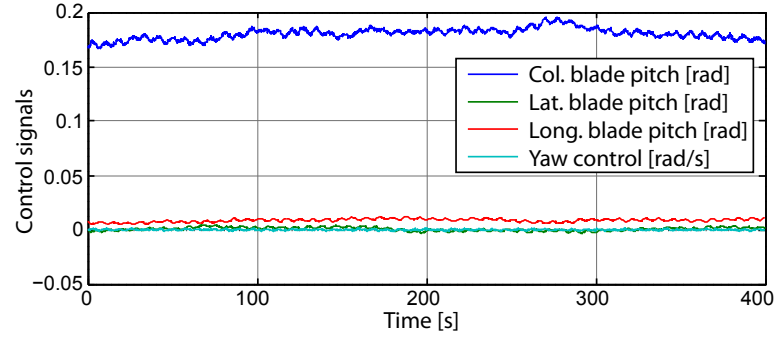


Figure 11.9: The 4 control signals for a period of 400 seconds.

in the performance plots in this section. The weights result in the controller named $\mathbf{K}_{flight,final[4 \times 12]}$ shown in (11.16) and included on the DVD in `SAH_flight.mat`. Note that the \mathbf{K} matrix is shown transposed.

$$\mathbf{K}_{flight,final}^T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -604.33 & 1.85 & -3.97 & -154.41 \\ -20.15 & -535.05 & -38.48 & -399.29 \\ 663.72 & 50.77 & -64.05 & 168.58 \\ -48.19 & -23.64 & -17.67 & 891.15 \\ -134.14 & -6.84 & 8.42 & 66.34 \\ 10.81 & -6.09 & -4.27 & -148.92 \\ -89.75 & 649.09 \cdot 10^{-3} & -1.88 & -56.62 \\ -1.16 & -1.24 & -202.31 \cdot 10^{-3} & -5.61 \\ 17.75 & 4.94 & -16.95 & -432.54 \\ -5.41 & -718.45 \cdot 10^{-3} & 524.46 \cdot 10^{-3} & 212.58 \end{bmatrix} \cdot 10^{-3} \quad (11.16)$$

The first two columns of the final flight controller has been defined to 0, since the flight controller should not control on the x- and y-positions at all, this is also the reason position turning is not necessary for the flight controller. The performance of the final flight controller with the estimated sensor measurements are used in the second acceptance test 13.1 on page 145.

11.2.4 Controller Switching Between Hover and Flight

A way of alternating between the hover and the flight controllers is wanted to make the system more autonomous. In this work this is by implementing a simple controller switching that switches between the two controllers using the distance from the helicopters position to the reference position as input to a z-shaped membership function.

The main blocks contained in the implementation of the controller shown in figure 11.10, include the helicopter yaw-offset for the flight controller explained in 11.2.3, a velocity scaling for the flight controller and the switching between the controllers. The position correction for the hover controller explained in 11.2.2 is also included.

Figure 11.11 show the helicopter velocity correction for the flight controller. This is a correction to the operating point for the x- and y-velocity of the flight controller, to have a smooth switching between the controllers. The correction is implemented as a saturation and rate limiter on the distance to the reference position seen from the helicopter, shown as the top part of figure 11.11. The lower part of the figure is a rotation of the x- and y-velocity operating points. This is implemented to make the helicopter fly towards the reference position, even though it is facing the wrong way. The rotation use cosine to the yaw error as x-velocity scaling, and sine to the yaw error for the y-velocity scaling.

The switching between the two controllers is a simple scaling of the controller gains, as shown in figure 11.12. The distance to the reference position seen from the helicopter is used as input for the scaling.

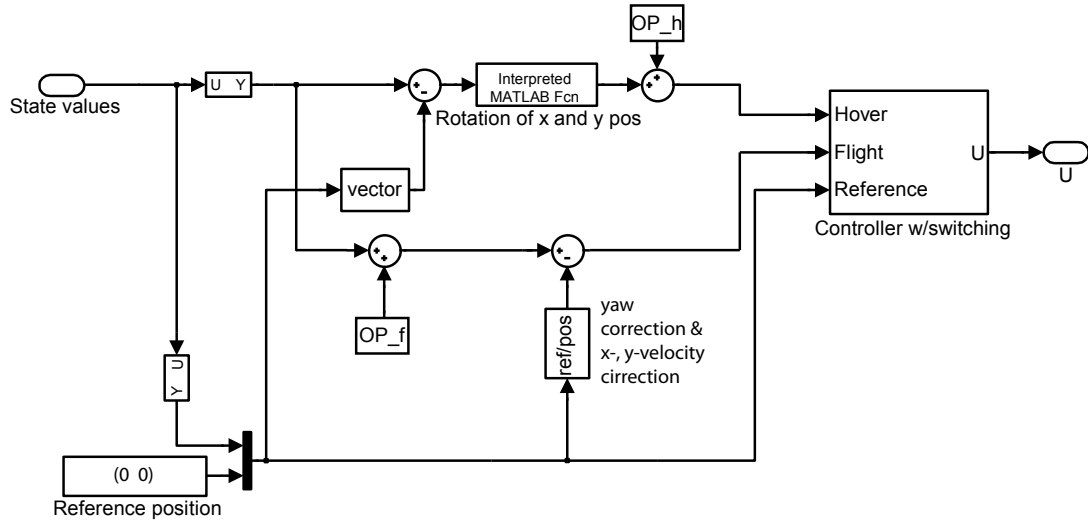


Figure 11.10: A diagram showing the main parts of the controller block.

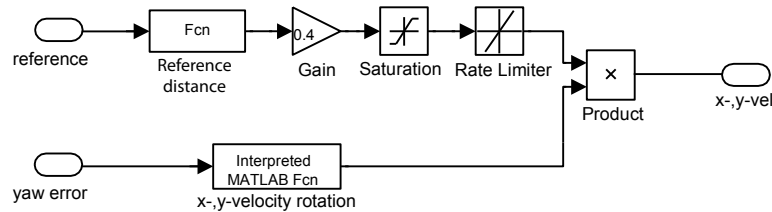


Figure 11.11: A diagram showing the helicopter x- and y-velocity correction, set in place to ensure the helicopter goes from hover to flight and from flight to hover in a smoother way than with velocity steps, aswell as flying in the right correction.

To test the performance of the two controllers with switching, a test where the helicopter is tasked with flying in a large square is set up. The corners of the square is given to the helicopter as reference points at a given time. The simulation starts with the helicopter hovering for 10 seconds, afterwards a reference position 100 meters away is given every 30 second, as the time window it has to get to the position and hover above it.

The x- and y-positions of the helicopter is shown in figure 11.13 where it is shown that the helicopter indeed reaches the reference points within the time window.

The combined control signals from the switching controller is shown in figure 11.14, where the control signals for the yaw control is shown separately, since it is much larger than the other control signals.

It is clear from the control signals when the reference position is changed, but also when the helicopter reaches its destination. While hovering, the helicopter is controlled to point in the direction of the x-axis, meaning that the helicopter has to turn around

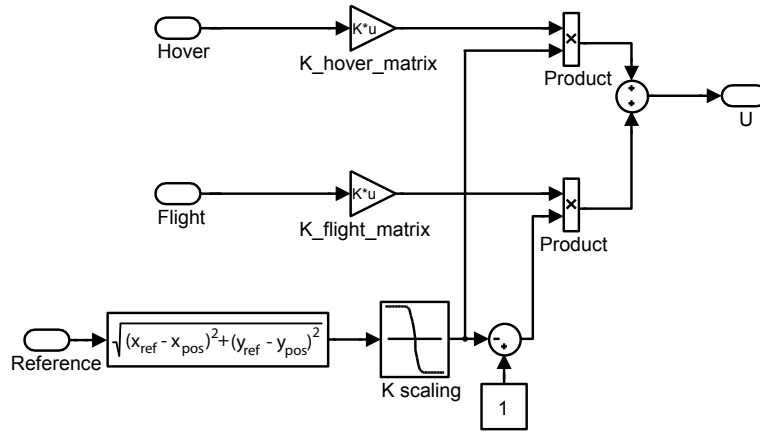


Figure 11.12: A diagram showing the controller switching block, using the distance to a reference position to manage the controller switching.

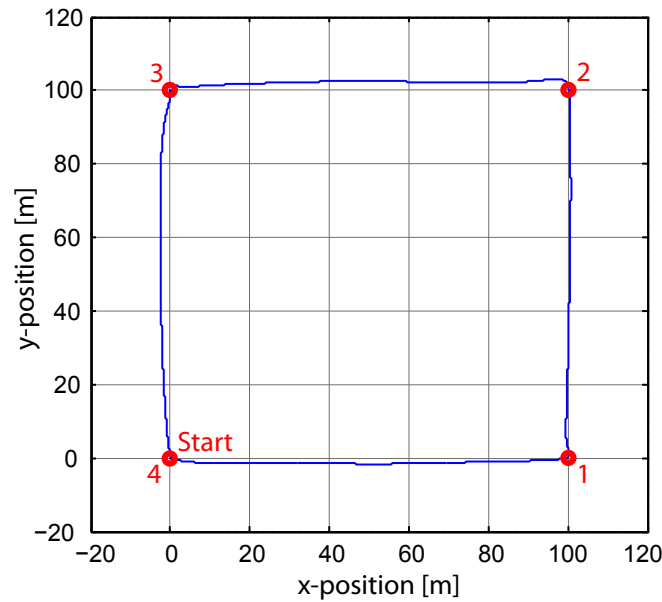


Figure 11.13: Plot of the x- and y-positions of the helicopter over the switching performance simulation.

when reference point 3 is given, and when that destination is reached. The resulting control signals are also the largest, as shown in the time period between 70 and 100 seconds.

Figure 11.15 show that the rate limiter on the x-velocity reference work as intended every time the helicopter goes from hover to flight and back, and that the velocity in between goes up to the reference of 6m/s, the negative velocity spikes that appear at the times where the helicopter approach the reference position could be a result of the hover

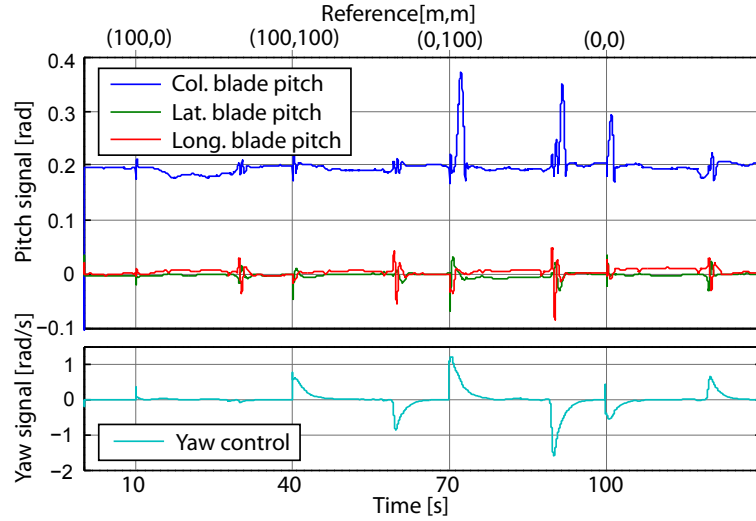


Figure 11.14: Control signals from the combined switching controller over the period of the switching performance simulation.

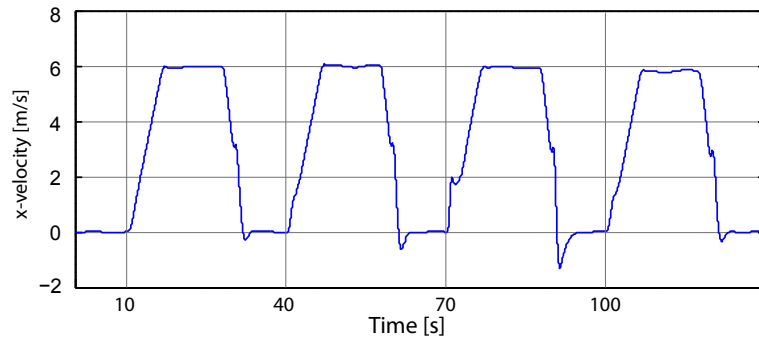


Figure 11.15: x-velocity seen from the helicopter over the switching performance simulation.

controller taking over, which is not subject to heavier weightings on the x-velocity, but rather the position of the helicopter.

11.3 LQ Control of a Helicopter with Slung Load

Here a helicopter with a slung load attached using a single wire is controlled using LQ control. Since the system dynamics has changed from the previous chapter, where no slung load is attached, a new linearisation, and a new pair of weight matrices has to be found. Just like the case without a slung load, two controllers will be derived in this section, one for hover and one for flight.

The initial weightings are found using perfect state feedback, meaning no sensor dynamics or noise are present in the tests in this section. Along with the weights on the states already presented in the standalone helicopter control section, the system now

also include states for the two Euler angles describing the position of the load relative to the helicopter, as well as the angular velocity of the two angles. Summed up, a total of 4 new states that need to be controlled.

11.3.1 Finding the Weight Matrices

The initial weight matrices for the helicopter and load controller, is a copy of the final weights, found in (11.12), with weights added on the new states, meaning that the \mathbf{Q} matrix now has 16 weights, while the \mathbf{R} matrix still has the same 4 weights.

$$\begin{aligned}\mathbf{Q}_{\text{diag,hover}} &= [1 \ 1 \ 20 \ 1.62 \ 1.62 \ 20 \ 1.62 \ 1.62 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 1.62 \ 1.62] \\ \mathbf{R}_{\text{diag,hover}} &= [50 \ 100 \ 100 \ 50]\end{aligned}\tag{11.17}$$

The 7th, 8th, 15th and 16th weights in the $\mathbf{Q}_{\text{diag,hover}}$ matrix are the new ones. They are the weightings of the load angles and load angular velocities respectively, and they are calculated from a max angle of $\pi/4$ rad and an angular velocity of $\pi/4$ rad/s.

The initial weightings for the flight controller is shown in (11.18), is also found by taking the weightings from the stand alone helicopter flight controller, and adding weights for the extra states.

$$\begin{aligned}\mathbf{Q}_{\text{diag,flight}} &= [10^{-10} \ 10^{-10} \ 10 \ 1 \ 1 \ 3 \ 0.40 \ 0.40 \ 1 \ 0.1 \ 0.1 \ 0.01 \ 0.01 \ 0.01 \ 1.91 \ 1.91] \\ \mathbf{R}_{\text{diag,flight}} &= [40 \ 800 \ 800 \ 5]\end{aligned}\tag{11.18}$$

The 7th and 8th weights in $\mathbf{Q}_{\text{diag,flight}}$ are chosen based on a maximum wanted deviation from the operating point by $\pi/2$ rad, since the angle is expected to be different from 0 when the load is being dragged. The 15th and 16th weights in $\mathbf{Q}_{\text{diag,flight}}$ are chosen based on a wanted maximum angular velocity of the load of $\pi/6$ rad/s, since the load should be kept steady.

With the weight matrices presented in (11.17) and (11.18), the next stage is to design controllers for flight and hover, based on these weights. The hover controller will be designed first in the following section.

11.3.2 Deriving a Controller for a Helicopter with Slung Load in Hover

For the controller design, full state feedback is assumed, same as it is for the previous controllers, meaning that the system can be expressed as shown on figure 11.1 on page 88. Before a controller can be created a linear system description is needed. Appendix B on page 179 presents the procedure used to create a linear state space model for the helicopter with slung load. Using the \mathbf{A} and \mathbf{B} matrices found by this procedure, together with the $\mathbf{Q}_{\text{diag,hover}}$ and $\mathbf{R}_{\text{diag,hover}}$ weight matrices found in (11.17), it is now possible to find a feedback matrix $\mathbf{K}_{\text{init,hover}}$ that minimises the performance function, given by (11.3).

The attained controller is able to keep the helicopter and load well within the desired limits, but both states and control signals oscillates at a frequency of about 2 Hz, which

in a real situation would cause a lot of stress, and potentially become unstable. This can be seen on figures 11.16, 11.17 and 11.18 on the next page, expressing the position of the helicopter, and the control signals respectively.

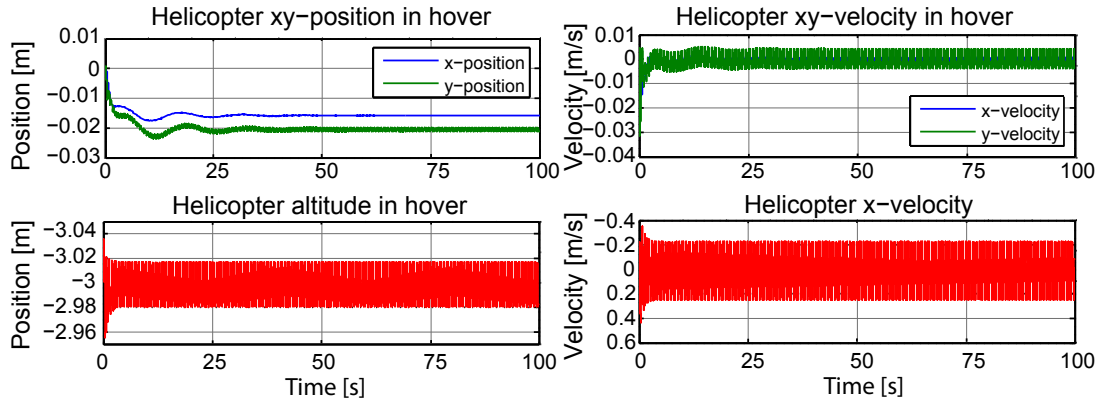


Figure 11.16: The top plots show the x- and y-positions and velocities of the helicopter. The bottom plot shows the altitude of the helicopter in the same scenario.

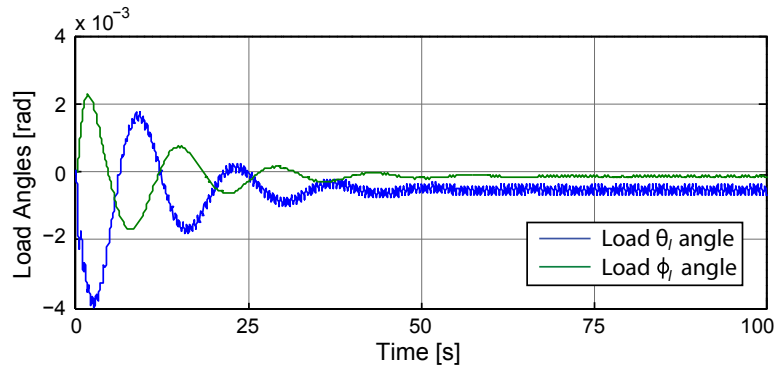


Figure 11.17: This plot shows the load angles when hovering with a controller constructed on the initial weights.

All states and more from this test, is included on the enclosed DVD, at the following path:



/Simulation/HAL/Initial_LQ/HAL_hover.mat

This file includes state space matrices, the control matrix Q and R weight matrices, as well as control signals, state- and initial values, for the simulation.

To slow down the dynamics and to gain more resilience towards this instability, the weights are tuned, which will lead to a less aggressive controller. The final weights for this hover controller is shown in (11.19).

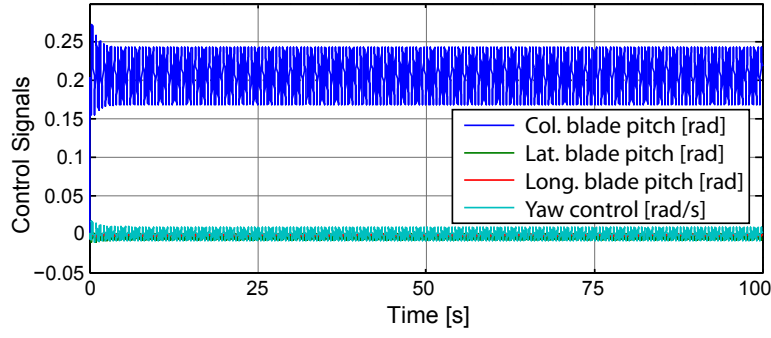


Figure 11.18: The plot shows the control signals to the helicopter. The control signals are small in amplitude, but with fast oscillations.

$$\begin{aligned} \mathbf{Q}_{\text{diag,hover}} &= [2 \ 2 \ 0.5 \ 1.62 \ 1.62 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0.5 \ 0.1 \ 0.1 \ 0.1 \ 1 \ 1] \\ \mathbf{R}_{\text{diag,hover}} &= [160 \ 330 \ 330 \ 3] \end{aligned} \quad (11.19)$$

The higher weights on the blade pitch control signals forces the controller to use smaller variations in the control signal, this reduces some of the oscillations, but the main reduction of oscillations originates from the reduced weight on the vertical position of the helicopter. To make the heading of the helicopter more steady, both the weight on the helicopter yaw angle and on the tail rotor angular velocity control signal is lowered greatly. The control matrix, $\mathbf{K}_{\text{final,hover}}$, attained with the previously mentioned state and input matrices and these new weights is shown in (11.20)

$$\mathbf{K}_{\text{final,hover}}^T = \begin{bmatrix} -1.94 & -37.56 & 68.15 & 19.95 \\ -6.42 & -67.69 & -37.15 & -93.25 \\ -230.85 & 21.27 & 7.15 & -307.67 \\ 4.49 & -187.80 & -114.97 & -80.70 \\ -2.04 & 133.88 & -283.12 & -123.77 \\ 51.02 & -11.92 & 231.01 \cdot 10^{-3} & 726.84 \\ 351.63 \cdot 10^{-3} & -32.36 & -17.96 & -54.21 \\ -7.73 & 18.35 & -30.52 & -90.48 \\ 138.14 \cdot 10^{-3} & -52.69 & 96.77 & 66.05 \\ -6.81 & -86.05 & -49.84 & -88.85 \\ -79.26 & 9.84 & 4.24 & -65.25 \\ 68.08 \cdot 10^{-3} & -8.28 & -1.71 & -3.44 \\ 312.20 \cdot 10^{-3} & 6.59 & -42.03 & -10.11 \\ 7.09 & -12.79 & -3.97 & 112.41 \\ -10.47 & 59.96 & 32.60 & 81.45 \\ 4.43 & -32.18 & 52.79 & 175.85 \end{bmatrix} \cdot 10^{-3} \quad (11.20)$$

Figure 11.19, 11.20 and 11.21 on page 102 shows the position of the helicopter, the load angles as well as the control signals, during a simulation with this new controller. Note that this simulation is performed with wind enabled. The wind used can be seen on figure 11.22 on page 102.

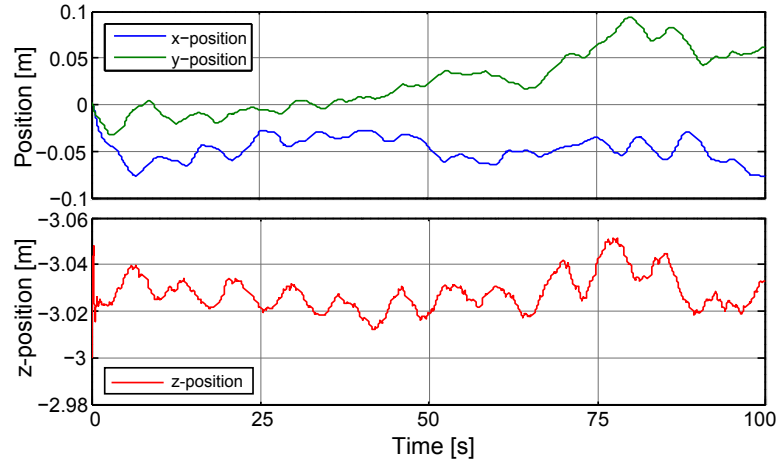


Figure 11.19: The top plot shows the x- and y-positions of the helicopter. The bottom plot shows the altitude of the helicopter in the same scenario.

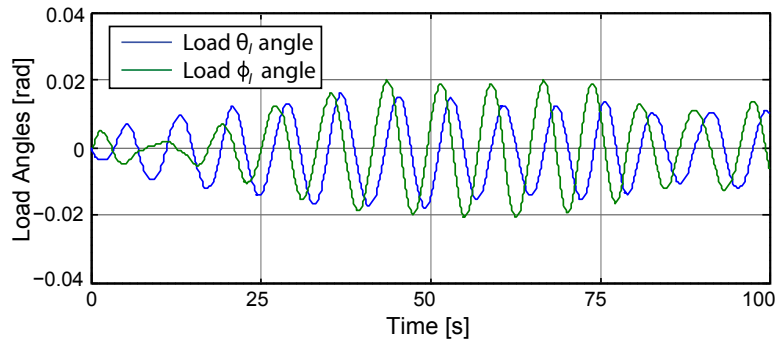


Figure 11.20: This plot shows the load angles when hovering with a controller constructed on the final weights.

The helicopter position plot show how the controller is able to keep the helicopter close the reference position at all time during the simulation. The altitude of the helicopter is above the reference at all time, which can be explained by the buoyancy of the slung load not being part of the trimming process. The load angles are shown to be small while hovering, with slow oscillations. The full result of a simulation with this controller, can be found on the enclosed DVD at the following path:



/Simulation/HAL/Final_LQ/HAL_hover.mat

The goal of the simulation is to hover at the point $\{0, 0, -3\}$ m. The simulation show a significant increase in the ability to hover steadily at a specific point, with much gentler control signals, even with wind enabled, with respect to the simulation with the initial hover controller.

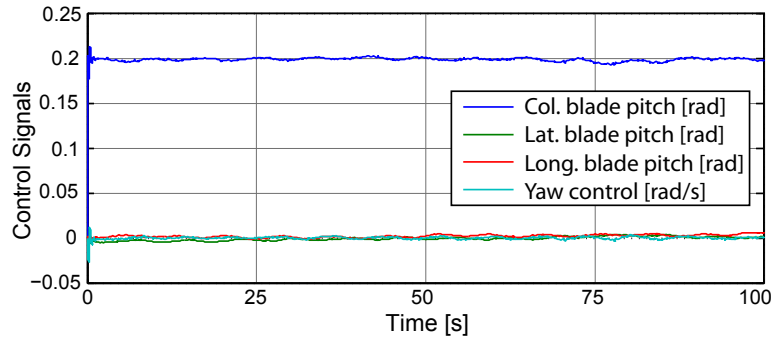


Figure 11.21: The plot shows the control signals to the helicopter when hovering.

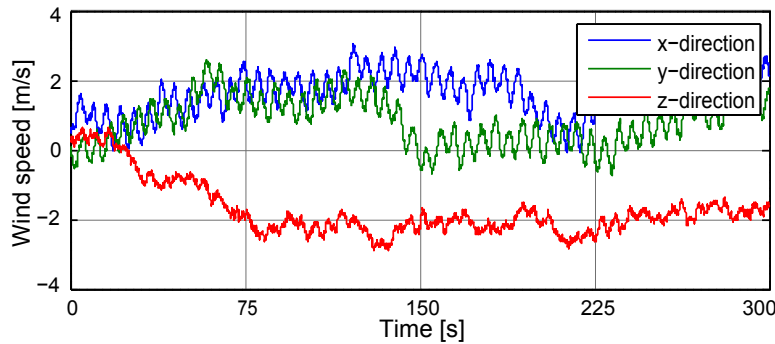


Figure 11.22: This graph shows the wind used for the test of $\mathbf{K}_{\text{final,hover}}$.

11.3.3 Deriving a Controller for a Helicopter with Slung Load in Flight

For the controller design for flight with slung load, full state feedback is again assumed, same as it is for the previous controllers. The \mathbf{A} and \mathbf{B} matrices for flight with load can be found with the methods presented in appendix B on page 179. Using these \mathbf{A} and \mathbf{B} matrices, together with the $\mathbf{Q}_{\text{diag,flight}}$ and $\mathbf{R}_{\text{diag,flight}}$ weight matrices found in (11.18) it is possible to find a feedback matrix $\mathbf{K}_{\text{init,flight}}$ that minimises the performance function, given by (11.3).

The produced controller is able to fly the helicopter at a fairly constant rate, with relatively calm control signals (although much too high at times), but is unable to maintain altitude, as it can be seen on figures 11.23 on the next page and 11.24 on the facing page.

The results for the simulations show how altitude of the helicopter fluctuate a lot while in flight, which could be explained by large variations in the collective blade pitch control signals. Further results from this simulations is available on the enclosed DVD at the following path:



/Simulation/HAL/Initial_LQ/HAL_flight.mat

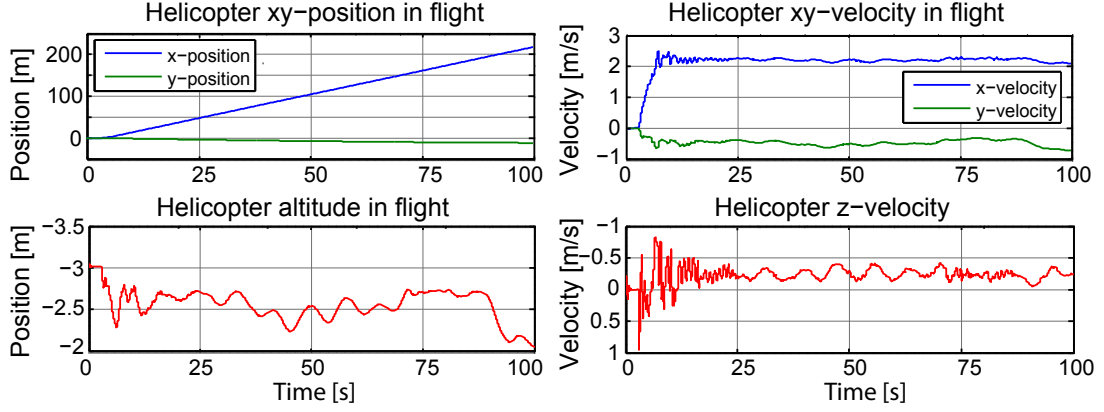


Figure 11.23: The top plots shows the x- and y-positions and velocities of the helicopter when flying. The bottom plots show the altitude of the helicopter and z-velocity in the same scenario.

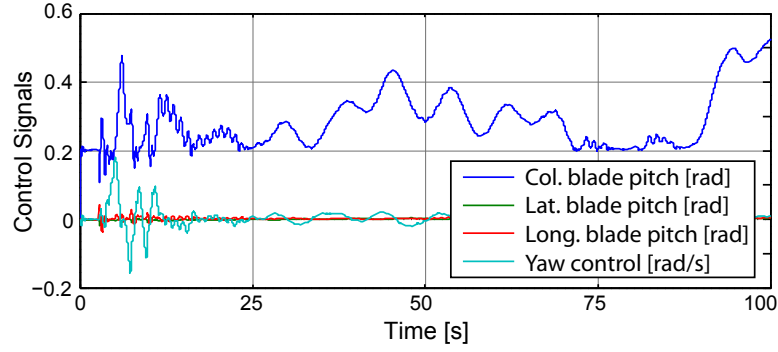


Figure 11.24: The plot shows the control signals to the helicopter when flying with initial flight control.

This file includes state space matrices, the control matrix Q and R weight matrices, as well as control signals, state- and initial values, for the simulation.

To avoid this loss of altitude and high control signals, the weights are tuned, which leads to a new controller. The final weights for this flight controller is shown in (11.21).

$$\begin{aligned} \mathbf{Q}_{\text{diag,flight}} &= [10^{-16} \ 10^{-16} \ 20 \ 2.62 \ 2.62 \ 4 \ 0 \ 0 \ 3 \ 3 \ 0.5 \ 0.1 \ 0.1 \ 0.1 \ 1 \ 1] \\ \mathbf{R}_{\text{diag,hover}} &= [260 \ 450 \ 450 \ 20] \end{aligned} \quad (11.21)$$

The tuned weights on the states are generally higher than the initial weights, except for the yaw angle and the load states. The yaw angle weight has been lowered, in order to prevent it from overshadowing other states. The weights on the load angles have been zeroed, and the weights on the load angular velocity have been lowered, to gain calmer helicopter movements. The weights on the control signals is now much closer to the control signal weights on the hover controller, which yields better results.

The control matrix, $\mathbf{K}_{\text{final,flight}}$, attained with the previously mentioned state and input matrices and these new weights is shown in (11.22).

$$\mathbf{K}_{\text{final,flight}}^T = \begin{bmatrix} 314.18 \cdot 10^{-6} & -46.97 \cdot 10^{-6} & -13.48 \cdot 10^{-6} & 490.26 \cdot 10^{-6} \\ 314.33 \cdot 10^{-6} & -46.86 \cdot 10^{-6} & -14.11 \cdot 10^{-6} & 491.59 \cdot 10^{-6} \\ -598.47 & -37.45 & 50.89 & -113.40 \\ 3.36 & -205.06 & -122.09 & -363.26 \\ 199.95 & 157.53 & -331.89 & -68.29 \\ -3.59 & -51.87 & -28.85 & 807.99 \\ 16.27 & -21.07 & -10.35 & 232.32 \\ -197.34 & -42.47 & 64.32 & 270.66 \\ -25.82 & -48.80 & 88.66 & 85.15 \\ -5.62 & -83.31 & -42.69 & -378.95 \\ -119.47 & 3.96 & -134.22 \cdot 10^{-3} & -59.33 \\ -62.90 \cdot 10^{-3} & -7.31 & -1.41 & -5.24 \\ 1.86 & 8.60 & -45.25 & -25.67 \\ -536.82 \cdot 10^{-3} & 375.88 \cdot 10^{-3} & 3.19 & 175.91 \\ 7.39 & -21.16 & -12.74 & 278.55 \\ -244.35 & -5.08 & -11.14 & 204.07 \end{bmatrix} \cdot 10^{-3} \quad (11.22)$$

A test where the helicopter is set to fly at a constant velocity in the x-direction is conducted and figure 11.25, shows the position and velocity of the helicopter during this test. It can be seen that the altitude of the helicopter is more constant and, and closer to the altitude reference of 2.8 meters, which is set to allow the slung load to be dragged behind the helicopter while in flight. It is again noteworthy that this test is conducted with wind enabled, whereas the one with the initial weightings is not.

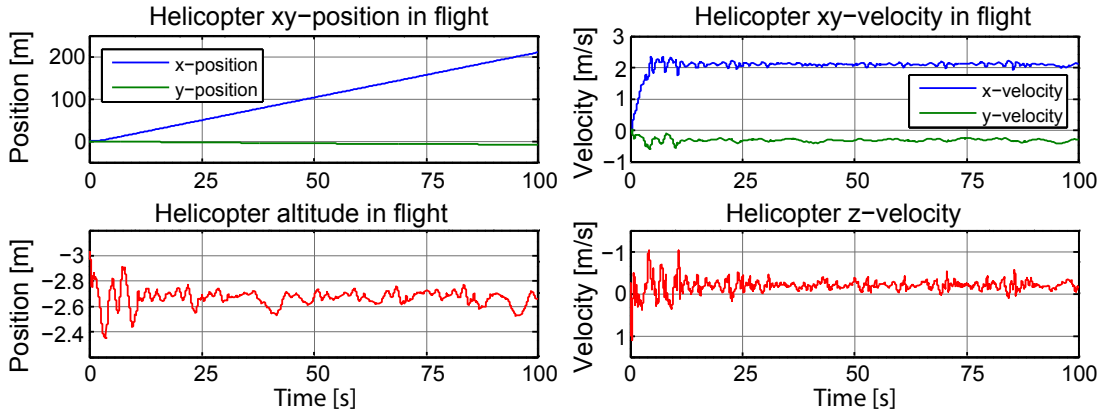


Figure 11.25: This figure shows the position and velocity of the helicopter during a flight simulation with a semi submerged load, at a reference velocity of 2m/s .

Figure 11.27 shows more variations in the control signals than before, but seems more contained to smaller deviations. Figure 11.26 show how the load states are close to constant after the startup, and keeping in mind that wind is enabled, the fast variations are explainable. The wind used in this test can be seen in figure 11.28 on the following page

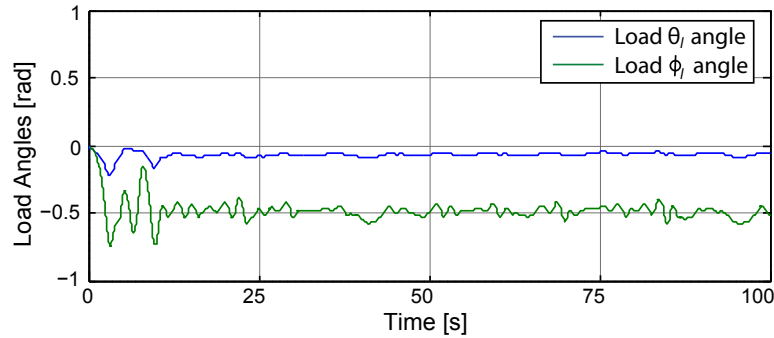


Figure 11.26: This plot shows the load angles when flying with a controller constructed on the final weights, given in (11.21).

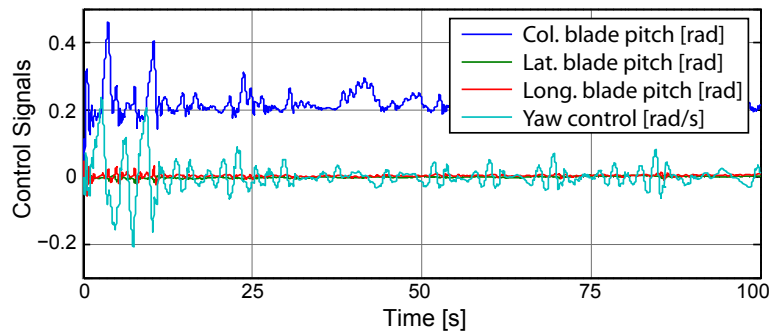


Figure 11.27: This figure shows the control signals to the helicopter during flight with a semi submerged load at 2m/s .

More results and data from this simulation is available at the enclosed DVD at the following path:



/Simulation/HAL/Final_LQ/HAL_flight.mat

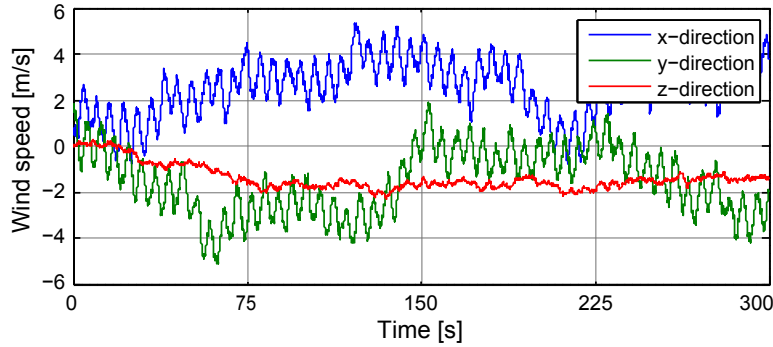


Figure 11.28: This figure shows the wind used for a simulation of flight with a semi submerged load, at a rate of 2m/s .

11.3.4 Controller Switching Between Hover and Flight

When flying with a load, the helicopter still needs to be able to switch between flight and hover, thus the same switching is made as for the load controllers as for the stand alone helicopter controllers. With this switching in place a simulation is conducted, where the helicopter is set to fly in a square, same as for the stand alone helicopter. However, since the helicopter is set to fly slower when flying with load, the time between reference changes is now 70 seconds, instead of 30. Figure 11.29 on the next page shows the xy-components of the helicopter position, and it can be seen that the helicopter flies between the specified points with a small error from the straight line connecting the points. This error occurs because the error in heading is relatively small when the helicopter is far from the target point.

By closer examination of the measured data, it can be seen that the helicopter reaches within 1 cm of each target point, before heading for the next one. The data supporting this can be found on the enclosed DVD at the following path:



/Simulation/HAL/Final_LQ/Square.mat

This file contains the state and input matrices, the flight and hover controllers, the flight and hover controller weights, the state values, initial values, control signals, and the wind, for the entire simulation.

Figure 11.30 on the facing page shows the forward and sideways velocity of the helicopter during this simulation. It shows that the controller is not entirely capable of keeping a forward velocity of 2m/s , and it has a significant contribution in sideways velocity. Keep in mind that the simulation is conducted with wind enabled, as shown by figure 11.32 on page 108.

That it is not quite capable of maintaining the forward velocity is also visible on the control signals where a collective pitch reference of more than 0.5rad can be observed.

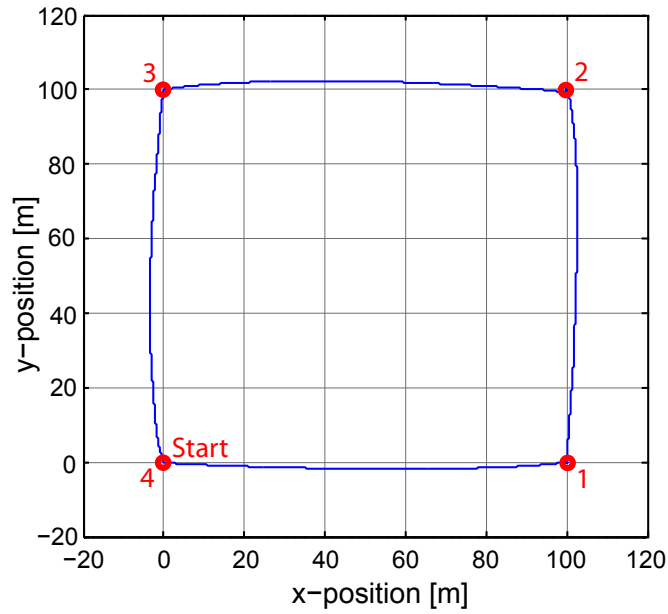


Figure 11.29: This plot shows the horizontal position of the helicopter when switching between flight and hover controllers while flying in a square pattern. A small deviation from the straight line between points can be observed in the trajectory.

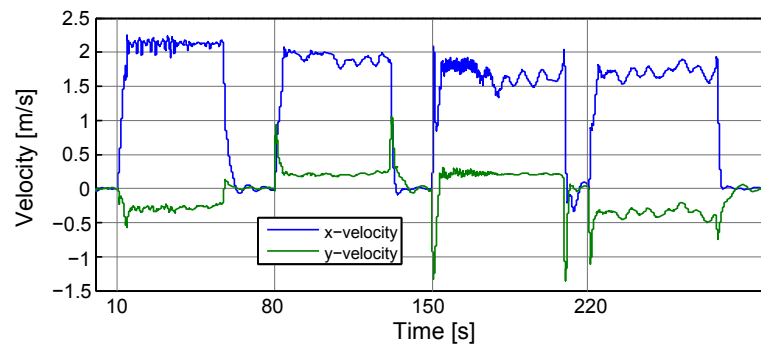


Figure 11.30: This plot shows the forward and sideways velocity of the helicopter during a simulation with switching between hover and flight.

This section shows the helicopters ability to fly with a semi submerged slung load, while switching between the designed LQ controllers for hover and flight.

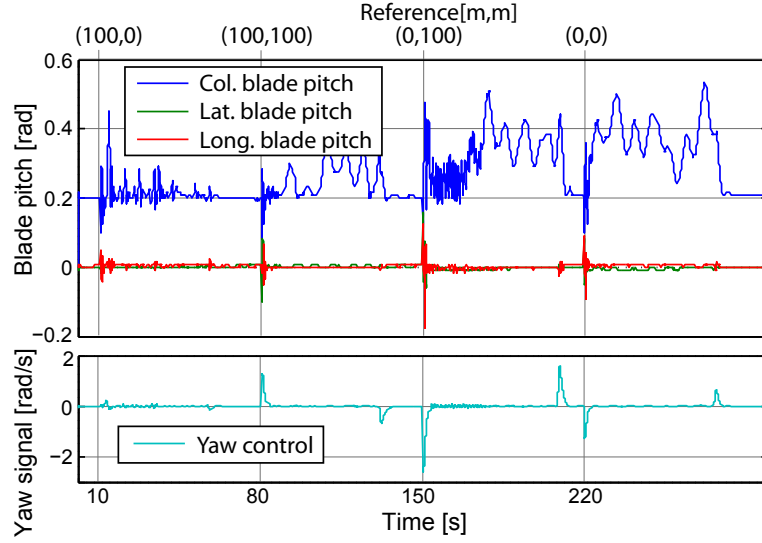


Figure 11.31: This plot shows the controller signals while flying with a semi-submerged load and switching between flight and hover controllers.

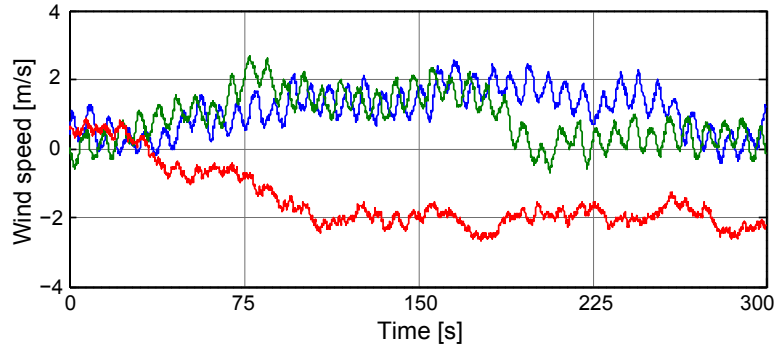


Figure 11.32: This plot shows the wind during this controller switching simulation.

11.4 LQ Controller Stability Regions

Figure 10.4 on page 80 and 10.5 on page 82 show the open loop poles of the standalone helicopter and helicopter with load respectively. These open loop poles are moved into the stable region by closing the loop with the produced controllers. However, as seen on the same figures, these poles move as the velocity of the helicopter changes. The purpose of this section is to investigate under which velocities stability can be guaranteed, with the produced controllers.

Stability Regions for the Stand Alone Helicopter

The model is linearised over a range of velocities from -30 to 30 m/s in steps of $1/10 \text{ m/s}$, with a side-slip angle of 0 to $\pi \text{ rad}$ in steps of $\pi/10 \text{ rad}$. A velocity is stable if all of the closed loop poles at that velocity has a strictly negative real part. Figure 11.33

shows the highest and lowest velocities that are stable for both the flight and hover controllers.

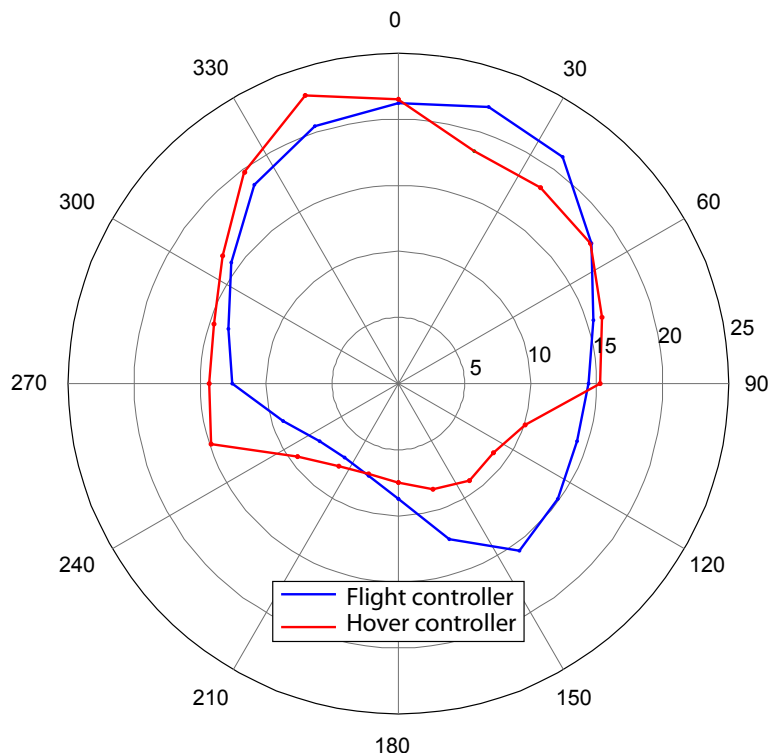


Figure 11.33: Stable velocity regions for the stand alone helicopter with flight and hover controllers, swept over velocity.

The figure shows that both controllers are better at handling high velocities in the forward direction than backwards. Both controllers ability to handle sideways velocities are in between their ability to handle forwards and backwards velocities. The regions are skewed, but to opposite sides, which is presumed to be caused by the variations in trimmed values, since the controllers are build for very specific operation points. It appears that the potential weak point of these two controllers is the flight controller, which when flying 6m/s in the 210° angle, comes close to the boundary.

Stability Regions for the Helicopter and Load

The model is again linearised over a range of velocities, this time from -6 to 6m/s in steps of $1/10\text{m/s}$, with a side-slip angle of 0 to π rad in steps of $\pi/10$ rad. A velocity is stable if all of the closed loop poles at that velocity has a strictly negative real part. Figure 11.34 on the following page shows the most extreme velocities that are stable for both the flight and hover controllers.

The two regions show heavily restricted stability regions, compared to the regions for the stand alone helicopter controllers. This is expected, since the open loop poles of the helicopter and load move far more, compared to the open loop poles of the stand alone helicopter. A noteworthy point is that the flight controller is not able to completely stabilise the system, when the helicopter is stationary, or moving outside the $\pm 60^\circ$

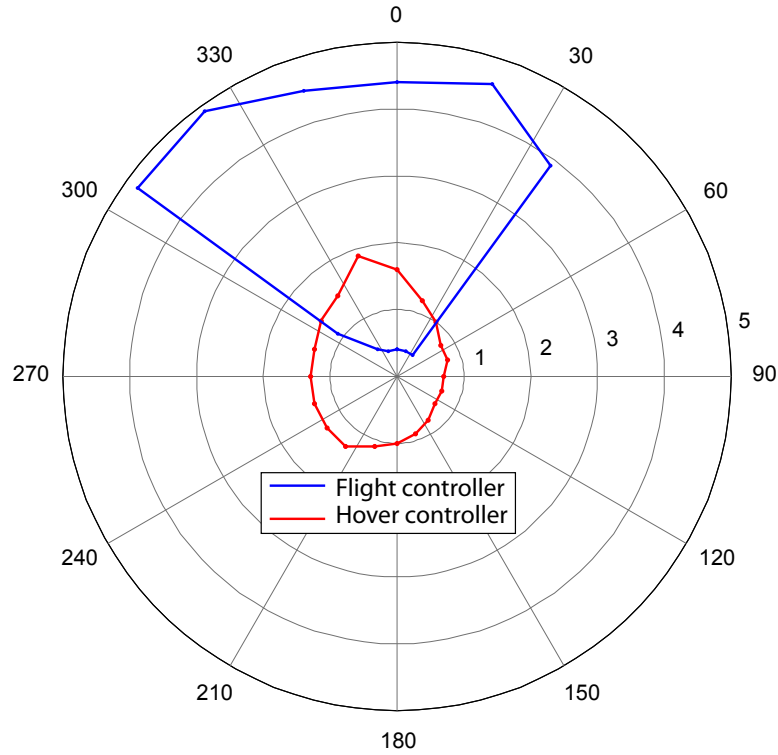


Figure 11.34: Stable velocity regions for the helicopter and load with flight and hover controllers, swept over velocity.

angle. The hover controller, even though not able to stabilise high velocities, shows a more uniform ability to stabilise velocities in any direction.

11.5 LQ Control Conclusion

A series of LQ controllers for a standalone helicopter and a helicopter with semi-submerged slung is produced in this chapter, including both hover and flight controllers for both system setups. The Bisgaard helicopter model is used for the standalone helicopter, whereas the helicopter with slung load include the slung load and rigid body modelling done in this work.

The results for the standalone helicopter controller, show that it is able to hover above a set-point position, and fly towards a new reference position. While in hover, the x- and y-position of the helicopter is used for control, where the flight controller use the x- and y-velocity of the helicopter, along with the yaw-angle, which is controlled to make the helicopter fly in the right direction, and have the correct heading while doing so.

When attaching a slung load to the helicopter, the priorities of the controller changes. While the states used for control is the same as for the standalone system, it is now much more relevant to keep the slung load steady in the water, and drag it smoothly,

than to reach the x- and y-velocity set-points while in flight.

Switching between hover and flight controllers is implemented so the helicopter is able to go from hover to flight and back to hover, which is what is needed in the rescue mission scenario description. Results for the switching show that the controllers with switching in both cases with and without slung load, is able to follow a preset pattern to an acceptable degree, the results also showed that some of the switching transactions is better than others, with the worst case results being when the helicopter had to turn around when going back to hover.

Model Predictive Control

The control strategy examined in this chapter is *Model Predictive Control* (MPC) also called *Receding Horizon Control* (RHC), which is a more advanced control strategy when compared to LQ and PID control, but also more computational heavy in most cases. The most common reason to use MPC is its ability to handle large sets of constraints in multi-variable systems, but also because of the prediction ability of MPC, where a set of future control signals and the resulting state values are estimated based on a system model.

This chapter is like the previous split up into a methodology section starting below, and a controller design section for the standalone helicopter, 12.2 on page 121, and helicopter with load, 12.3 on page 129. The theory introduced in the first section of this chapter is largely based on [16] and [18].

12.1 Model Predictive Control Methodology

MPC is as mentioned in the introduction often a very powerful control strategy when dealing with multi-state systems both with and without constraints, which is also why it widely used in the process control industry[8]. The general idea of MPC is shown in figure 12.1 on the next page, where a single state and control signal of an example system is plotted.

Discrete time is assumed with k denoting the current discrete time step, $\mathbf{y}(k)$ the current state measurement and $\mathbf{u}(k)$ the calculated control signal. The predictive controller also have a set-point trajectory which is a predefined trajectory for the state, and a reference trajectory that starts at the current state measurement, which is calculated as the path for the controller to follow back to the set-point trajectory.

The shape of the reference trajectory is predetermined, most often it is chosen to be an exponential function as the one shown in figure 12.1 as $\mathbf{r}(t|k)$. Other shapes for the reference trajectory can also be chosen such as an straight line, in this work however it is defined as an exponential function as shown in (12.1).

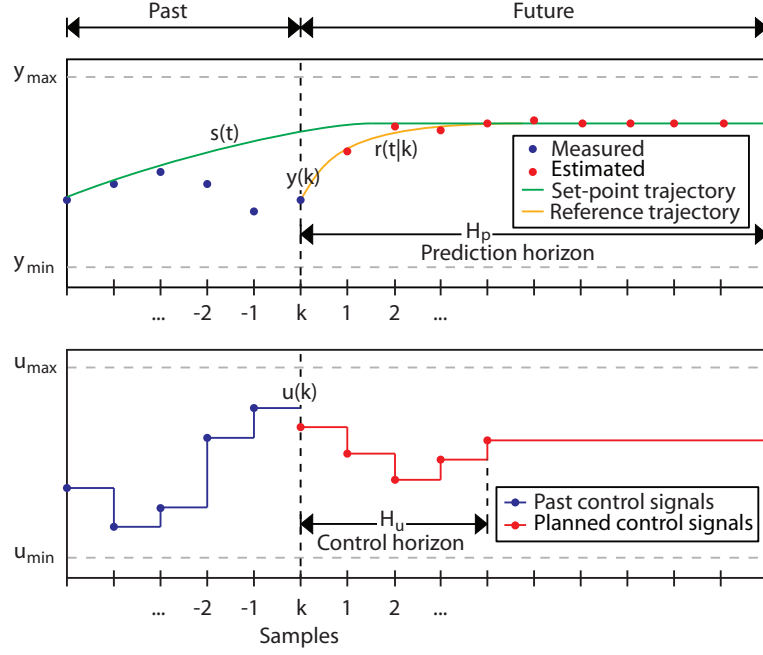


Figure 12.1: The general idea of model predictive control, including control horizon, prediction horizon, set-point and reference trajectory.

$$\begin{aligned}
 \epsilon(k) &= s(k) - y(k) \\
 \epsilon(k+i) &= e^{-iT_s/T_{ref}} \epsilon(k) \\
 r(k+i|k) &= s(k+i) - \epsilon(k+i)
 \end{aligned} \tag{12.1}$$

$s(k)$	The predefined set-point trajectory.
$\epsilon(k)$	The error between the set point trajectory and the measurement.
$r(k+i k)$	Reference trajectory, recalculated at each new time step.
T_{ref}	"Time constant", defining the speed of the reference trajectory response.

The predictive controller also has an *internal model* of the system, so it is able to predict the behaviour of the system and the control signals needed to return the system to the set-point trajectory at the current and future time step, specified by the *prediction horizon*. The internal model is often a linear model of the system, since MPC is not always possible on non-linear systems due to the optimisation problem, which is the core of model predictive control, where the optimal control signals used to return the plant to the set-point trajectory is calculated by minimising a cost function. Also the computational time is assured to be longer with a nonlinear model, which is not desired. In this work, a linear model is used for the predictive control.

With the future system responses and input trajectory calculated, the first step of the trajectory is applied to the system as control signal, and the cycle with state measure-

ment, system prediction and input trajectory is started again. At the new step the prediction horizon is along with all the other steps shifted one time step, which is the reason Model Predictive control also is known as Receding Horizon Control in some literature.

Due to the complexity of MPC, and the computational time that follows, MPC is often implemented on top of a lower level control loop containing a PID or LQ controller, which can work much faster, and then have the slower MPC correcting the set-points for the lower level controllers. This approach is mostly used on unstable systems where the lower level controller is introduced to stabilise the system.

12.1.1 Predictive Control Problem Formulation and Preliminaries

The linear system model is the discrete version of the system model used in LQ control chapter 11, on the form shown in (12.2), with the only change from a normal discrete state space representation is the possibility of only using predictive control on some of the plant outputs through the constrained performance signals $\mathbf{z}(k)$.

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_y\mathbf{x}(k) \quad \mathbf{z}(k) = \mathbf{C}_z\mathbf{x}(k)\end{aligned}\tag{12.2}$$

Here assuming there is no feed-forward dynamics $\mathbf{D}_{z,y}$, meaning that $\mathbf{y}(k)$ and $\mathbf{z}(k)$ does not depend directly on the current input $\mathbf{u}(k)$. In cases where all of the outputs in $\mathbf{y}(k)$ needs to be controlled $\mathbf{z}(k) \equiv \mathbf{y}(k)$. It should be noted that in the remaining part of the chapter, predicted values such as $\hat{\mathbf{u}}(k+i|k)$ is written as $\hat{\mathbf{u}}(k+i)$ unless the prediction is made based on another time step than k .

The system model shown in (12.2) is however often constrained by upper and lower limits of some or all state values, control signals, or the change rate of control signals. All these constraints contained in matrices \mathbf{E} , \mathbf{F} and \mathbf{G} , in the form showed in (12.3), defined as in (12.5). The constraints can be defined for all control signals and changes in $\hat{\mathbf{u}}(k+i)$ and $\Delta\hat{\mathbf{u}}(k+i)$ up until the control horizon H_u , and for the predicted state values in $\hat{\mathbf{z}}(k+i+1)$ up until the prediction horizon H_p . Note that $\Delta\mathbf{u}$, \mathbf{u} and \mathbf{z} are column vectors consisting of column vectors $\Delta\hat{\mathbf{u}}(k+i)$, $\hat{\mathbf{u}}(k+i)$ and $\hat{\mathbf{z}}(k+i+1)$.

$$\mathbf{E} \begin{bmatrix} \Delta\mathbf{u}(k) \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{F} \begin{bmatrix} \mathbf{u}(k) \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{G} \begin{bmatrix} \mathbf{z}(k) \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix}\tag{12.3}$$

$$\Delta\mathbf{u}(k) = \begin{bmatrix} \Delta\hat{\mathbf{u}}(k) \\ \vdots \\ \Delta\hat{\mathbf{u}}(k+H_u-1) \end{bmatrix} \quad \mathbf{u}(k) = \begin{bmatrix} \hat{\mathbf{u}}(k) \\ \vdots \\ \hat{\mathbf{u}}(k+H_u-1) \end{bmatrix} \quad \mathbf{z}(k) = \begin{bmatrix} \hat{\mathbf{z}}(k+1) \\ \vdots \\ \hat{\mathbf{z}}(k+H_p) \end{bmatrix}\tag{12.4}$$

E	Upper and lower constraints for changes in control signals.
F	Upper and lower constraints for control signals.
G	Upper and lower constraints for system outputs to be controlled.
$\hat{\mathbf{u}}(k+i)$	Predicted control signal at time $(k+i)$.
$\Delta\hat{\mathbf{u}}(k+i)$	Predicted change in control signal from time $(k+i-1)$ to $(k+i)$.
$\hat{\mathbf{z}}(k+i+1)$	Predicted performance signals/State values given $\hat{\mathbf{u}}(k+i)$.
H_p	Prediction horizon.
H_u	Control horizon.

With **E**, **F** and **G** being matrices including all the upper and lower constraints of respectively $\Delta\mathbf{u}$, \mathbf{u} and \mathbf{z} . The constraint matrices grows proportional to the length of the control and prediction horizons and grow large at long horizons, the assembly of the matrices are however often automated by software implementation, as with the MPC toolbox in MATLAB. An example of how the constraint matrices are assembled, is shown in (12.5), where the prediction horizon $H_p = 1$ and only a single control signal with constraints is present.

$$\begin{array}{c}
 u_{min} \leq u \qquad \qquad \qquad u_{max} \geq u \\
 \left\{ \begin{array}{l} 1 \leq \frac{1}{u_{min}}u \quad for \quad u_{min} > 0 \\ 0 \leq u \quad for \quad u_{min} = 0 \\ 1 \geq \frac{1}{u_{min}}u \quad for \quad u_{min} < 0 \end{array} \right. \quad \left\{ \begin{array}{l} 1 \geq \frac{1}{u_{max}}u \quad for \quad u_{max} > 0 \\ 0 \geq u \quad for \quad u_{max} = 0 \\ 1 \leq \frac{1}{u_{max}}u \quad for \quad u_{max} < 0 \end{array} \right. \\
 \left\{ \begin{array}{l} 0 \geq -\frac{1}{u_{min}}u + 1 \\ 0 \geq -u \\ 0 \geq \frac{1}{u_{min}}u - 1 \end{array} \right. \quad \left\{ \begin{array}{l} 0 \geq \frac{1}{u_{max}}u - 1 \\ 0 \geq u \\ 0 \geq -\frac{1}{u_{max}}u + 1 \end{array} \right.
 \end{array} \quad (12.5)$$

Put into tensor form this amounts to:

$$\underbrace{\left[\begin{array}{cc} \left\{ \begin{array}{l} -\frac{1}{u_{min}} \quad for \quad u_{min} \neq 0 \\ -1 \quad for \quad u_{min} = 0 \end{array} \right. & \text{sgn}(u_{min}) \\ \left\{ \begin{array}{l} \frac{1}{u_{max}} \quad for \quad u_{max} \neq 0 \\ 1 \quad for \quad u_{max} = 0 \end{array} \right. & -\text{sgn}(u_{max}) \end{array} \right]}_{\mathbf{F}} \begin{bmatrix} \hat{u}(k) \\ 1 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (12.6)$$

Note that $u(k)$ has been switched by $\hat{u}(k)$, this is because at the time when this equality is needed, $u(k)$ is not yet available.

The $\Delta\hat{\mathbf{u}}(k+i)$ solution that minimises the cost function $\mathcal{V}(k)$, which is presented in (12.7), is said to be optimal, and is the solution where the first step is applied on the system. If the solution is bounded by the constraints, the solution is no longer optimal, but the best possible within the boundaries.

$$\mathcal{V}(k) = \sum_{i=1}^{H_p} \| \hat{\mathbf{z}}(k+i) - \mathbf{r}(k+i) \|_{\mathbf{Q}(i)}^2 + \sum_{i=0}^{H_u-1} \| \Delta\hat{\mathbf{u}}(k+i) \|_{\mathbf{R}(i)}^2 \quad (12.7)$$

$\mathbf{r}(k+i)$	Reference trajectory at time $k+i$.
$\mathbf{Q}(i)$	State weight matrix, may vary over H_p .
$\mathbf{R}(i)$	Control signal weight matrix, may vary over H_p .

The notation $\| \mathbf{x} \|_{\mathbf{Q}}^2$ is a compact representation of a quadratic function, which when written out would look like $\mathbf{x}^T \mathbf{Q} \mathbf{x}$.

12.1.2 Prediction of Future States and Outputs

With the cost function and state space model introduced, it is time to compute the prediction steps, where the simplest case is full state measurements, without any knowledge about the measurement disturbances. In that case all there is to do is use the given system model to predict response through iteration.

The state values is predicted as far into the future as the prediction horizon H_p dictates. However, the computed control signals also change up until a point dictated by the control horizon H_u , where $H_u \leq H_p$, so the input changes in the interval $k, k+1, \dots, k+H_u$ and is therefore constant in the remaining range from H_u to $H_p - 1$. Based on the model and the control signals, the prediction of the future states can be found as in (12.8).

$$\begin{aligned}
 \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}\hat{\mathbf{u}}(k) \\
 \hat{\mathbf{x}}(k+2) &= \mathbf{A}\hat{\mathbf{x}}(k+1) + \mathbf{B}\hat{\mathbf{u}}(k+1) \\
 &= \mathbf{A}^2\hat{\mathbf{x}}(k) + \mathbf{A}\mathbf{B}\hat{\mathbf{u}}(k) + \mathbf{B}\hat{\mathbf{u}}(k+1) \\
 &\vdots \\
 \hat{\mathbf{x}}(k+H_p) &= \mathbf{A}^{H_p}\hat{\mathbf{x}}(k) + \mathbf{A}^{H_p-1}\mathbf{B}\hat{\mathbf{u}}(k) + \dots + \mathbf{B}\hat{\mathbf{u}}(k+H_p-1) \quad (12.8)
 \end{aligned}$$

This formulation where the predicted state is expressed from $\hat{\mathbf{u}}(k+i)$ can become difficult to optimise in the presence of constraints, where a formulation expressed from terms of $\Delta\hat{\mathbf{u}}(k+i)$ rather than $\hat{\mathbf{u}}(k+i)$ would result in a *Quadratic Programming* problem, which is a well known convex optimisation problem, with several solution methods and tools available. Having the definition of $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$ inserted into (12.8), while having in mind that the control signals only change up until H_u , the new expression of $\hat{\mathbf{x}}(k+i)$ is shown in (12.9). And afterwards set on vector form in (12.10).

$$\begin{aligned}
 \hat{\mathbf{x}}(k+1) &= \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}(\Delta\hat{\mathbf{u}}(k) + \hat{\mathbf{u}}(k-1)) \\
 \hat{\mathbf{x}}(k+2) &= \mathbf{A}^2\hat{\mathbf{x}}(k) + (\mathbf{A}\mathbf{B} + \mathbf{B})\hat{\mathbf{u}}(k-1) + \mathbf{A}\mathbf{B}\Delta\hat{\mathbf{u}}(k) + \mathbf{B}\Delta\hat{\mathbf{u}}(k+1) \\
 &\vdots \\
 \hat{\mathbf{x}}(k+H_p) &= \mathbf{A}^{H_p}\hat{\mathbf{x}}(k) + \sum_{i=1}^{H_p-1} \mathbf{A}^i\mathbf{B}\mathbf{u}(k-1) \\
 &\quad + \sum_{i=0}^{H_p-1} \mathbf{A}^i\mathbf{B}\Delta\mathbf{u}(k) + \dots + \sum_{i=0}^{H_p-H_u} \mathbf{A}^i\mathbf{B}\Delta\hat{\mathbf{u}}(k+H_u-1) \quad (12.9) \\
 &\Downarrow
 \end{aligned}$$

$$\begin{aligned}
\underbrace{\begin{bmatrix} \hat{\mathbf{x}}(k+1) \\ \vdots \\ \hat{\mathbf{x}}(k+H_u) \\ \hat{\mathbf{x}}(k+H_u+1) \\ \vdots \\ \hat{\mathbf{x}}(k+H_p) \end{bmatrix}}_{\mathcal{X}} &= \underbrace{\begin{bmatrix} \mathbf{A} \\ \vdots \\ \mathbf{A}^{H_u} \\ \mathbf{A}^{H_u+1} \\ \vdots \\ \mathbf{A}^{H_p} \end{bmatrix}}_{\mathcal{A}} \mathbf{x}(k) + \underbrace{\begin{bmatrix} \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{A}^i \mathbf{B} \\ \sum_{i=0}^{H_u} \mathbf{A}^i \mathbf{B} \\ \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{A}^i \mathbf{B} \end{bmatrix}}_{\mathcal{B}_u} \mathbf{u}(k-1) \\
&+ \underbrace{\begin{bmatrix} \mathbf{B} & \cdots & 0 \\ \mathbf{AB} + \mathbf{B} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_u-1} \mathbf{A}^i \mathbf{B} & \cdots & \mathbf{B} \\ \sum_{i=0}^{H_u} \mathbf{A}^i \mathbf{B} & \cdots & \mathbf{AB} + \mathbf{B} \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{H_p-1} \mathbf{A}^i \mathbf{B} & \cdots & \sum_{i=0}^{H_p-H_u} \mathbf{A}^i \mathbf{B} \end{bmatrix}}_{\mathcal{B}_{\Delta u}} \underbrace{\begin{bmatrix} \Delta \hat{\mathbf{u}}(k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+H_u-1) \end{bmatrix}}_{\Delta \mathcal{U}} \quad (12.10)
\end{aligned}$$

Having the predicted state values expressed in form of (12.10), the predictions of the output $\hat{\mathbf{z}}(k+i)$ can now be expressed as in (12.11) and in matrix form in (12.12).

$$\begin{aligned}
\hat{\mathbf{z}}(k+1) &= \mathbf{C}_z \hat{\mathbf{x}}(k+1) \\
\hat{\mathbf{z}}(k+2) &= \mathbf{C}_z \hat{\mathbf{x}}(k+2) \\
&\vdots \\
\hat{\mathbf{z}}(k+H_p) &= \mathbf{C}_z \hat{\mathbf{x}}(k+H_p) \quad (12.11) \\
&\Downarrow
\end{aligned}$$

$$\underbrace{\begin{bmatrix} \hat{\mathbf{z}}(k+1) \\ \vdots \\ \hat{\mathbf{z}}(k+H_p) \end{bmatrix}}_{\mathcal{Z}} = \underbrace{\begin{bmatrix} \mathbf{C}_z & 0 & \cdots & 0 \\ 0 & \mathbf{C}_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{C}_z \end{bmatrix}}_{\mathcal{C}} \underbrace{\begin{bmatrix} \hat{\mathbf{x}}(k+1) \\ \vdots \\ \hat{\mathbf{x}}(k+H_p) \end{bmatrix}}_{\mathcal{X}} \quad (12.12)$$

12.1.3 Solving the Predictive Control Problem

To find a solution, some new definitions have to be made. While having already defined $\mathcal{U}(k)$ and $\mathcal{Z}(k)$ in (12.4), and $\mathcal{T}(k)$ defined as in (12.13). The cost function presented in (12.7) can be rewritten to (12.14)

$$\mathcal{T}(k) = \begin{bmatrix} \hat{\mathbf{r}}(k+1) \\ \vdots \\ \hat{\mathbf{r}}(k+H_p) \end{bmatrix} \quad (12.13)$$

$$\mathcal{V}(k) = \| \mathcal{Z}(k) - \mathcal{T}(k) \|_{\mathcal{Q}}^2 + \| \Delta \mathcal{U}(k) \|_{\mathcal{R}}^2 \quad (12.14)$$

With \mathbf{Q} and \mathbf{R} including the weights from all the states and control signals in the range respectively (k) to $(k + H_p)$ and (k) to $(k + H_u - 1)$ as shown in (12.15). One of the reasons that the weight matrices \mathbf{Q} and \mathbf{R} changes over the prediction horizon is that the predicted values is less trusted along the prediction horizon, and thus not weighted equal to the present values.

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}(1) & 0 & \cdots & 0 \\ 0 & \mathbf{Q}(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{Q}(H_p) \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}(0) & 0 & \cdots & 0 \\ 0 & \mathbf{R}(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{R}(H_u - 1) \end{bmatrix} \quad (12.15)$$

If the weight matrices $\mathbf{Q}(k + i)$ and $\mathbf{R}(k + i)$ are diagonal matrices, the resulting \mathbf{Q} and \mathbf{R} matrices will also be diagonal matrices.

\mathcal{T} is already a function of the predicted control signals $\Delta\mathbf{U}$ which is the solution that minimise the cost function, so \mathcal{Z} is needed as a function of $\Delta\mathbf{U}$. In (12.11) with $\hat{\mathbf{x}}(k + i)$ inserted from (12.9), \mathcal{Z} has the form shown in (12.16).

$$\mathcal{Z}(k) = \Psi\hat{\mathbf{x}}(k) + \Upsilon\mathbf{u}(k - 1) + \Theta\Delta\mathbf{U}(k) \quad (12.16)$$

Ψ	Equal to \mathcal{CA} , defined in (12.10) and (12.12).
Υ	Equal to \mathcal{CB}_u , defined in (12.10) and (12.12).
Θ	Equal to $\mathcal{CB}_{\Delta u}$, defined in (12.10) and (12.12).

And finally defining a tracking error $\mathcal{E}(k)$ defined as the difference between the target trajectory and the response the system would have if no changes are made to the control signals as shown in (12.17).

$$\mathcal{E}(k) = \mathcal{T}(k) - \Psi\hat{\mathbf{x}}(k) - \Upsilon\mathbf{u}(k - 1) \quad (12.17)$$

While having the newly rewritten cost function \mathcal{V} , with the tracking error defined \mathcal{E} , the cost function can now be rewritten to terms that are dependent on $\Delta\mathbf{U}$, and those that are not, as shown in (12.18).

$$\begin{aligned} \mathcal{V}(k) &= \|\mathcal{Z}(k) - \mathcal{T}(k)\|_{\mathbf{Q}}^2 + \|\Delta\mathbf{U}(k)\|_{\mathbf{R}}^2 \\ \mathcal{V}(k) &= \mathcal{E}(k)^T \mathbf{Q} \mathcal{E}(k) - 2\Delta\mathbf{U}(k)^T \Theta^T \mathbf{Q} \mathcal{E}(k) + \Delta\mathbf{U}(k)^T [\Theta \mathbf{Q} \Theta + \mathbf{R}] \Delta\mathbf{U}(k) \\ \mathcal{V}(k) &= \text{constant} - \Delta\mathbf{U}(k) \mathcal{G} + \Delta\mathbf{U}(k)^T \mathcal{H} \Delta\mathbf{U} \end{aligned} \quad (12.18)$$

With \mathcal{G} and \mathcal{H} defined as in (12.19).

$$\mathcal{G} = 2\Theta^T \mathbf{Q} \mathcal{E}(k) \quad \mathcal{H} = \Theta^T \mathbf{Q} \Theta + \mathbf{R} \quad (12.19)$$

Where neither \mathcal{G} nor \mathcal{H} depends on $\Delta\mathbf{u}(k)$, so to find the optimal $\Delta\mathbf{u}$, the gradient of \mathcal{V} can be found and set to zero, as shown in (12.20). The result from this is the optimal control signals in all the time steps of the control horizon, which means the control signals for time k have to be picked out, since only those are applied to the system, shown in (12.21).

$$\nabla_{\Delta\mathbf{u}(k)} = -\mathcal{G} + 2\mathcal{H}\Delta\mathbf{u}(k) = 0 \quad \implies \quad \Delta\mathbf{u}(k)_{opt} = \frac{1}{2}\mathcal{H}^{-1}\mathcal{G} \quad (12.20)$$

$$\Delta\mathbf{u}(k)_{opt} = [I_l \ 0_l \ \cdots \ 0_l] \Delta\mathbf{u}(k)_{opt} \quad (12.21)$$

Keep in mind that the equations and solutions shown up until now is only for an unconstrained case, with all states measured where nothing is known about the measurement disturbances. To give an overview of the sizes of the matrices and vectors introduced in this section, table 12.1 is introduced, with the plant having l control signals, n states and m control outputs.

Name	Dimensions	Description
\mathbf{u}	$(l \ H_u) \times 1$	Predicted control signals over H_u .
$\Delta\mathbf{u}$	$(l \ H_u) \times 1$	Predicted control signal changes over H_u .
\mathcal{X}	$(n \ H_p) \times 1$	Predicted state values over H_p .
\mathcal{Z}	$(m \ H_p) \times 1$	Predicted outputs over H_p .
\mathcal{A}	$(n \ H_p) \times n$	Defined in (12.10).
\mathcal{B}_u	$(n \ H_p) \times l$	Defined in (12.10).
$\mathcal{B}_{\Delta u}$	$(n \ H_p) \times (l \ H_u)$	Defined in (12.10).
\mathcal{Q}	$(m \ H_p) \times (m \ H_p)$	Weightings for control outputs over H_p .
\mathcal{R}	$(l \ H_u) \times (l \ H_u)$	Weightings for control signals over H_u .
Ψ	$(m \ H_p) \times n$	Equal to $\mathcal{C}\mathcal{A}$, defined in (12.10) and (12.12).
Υ	$(m \ H_p) \times l$	Equal to $\mathcal{C}\mathcal{B}_u$, defined in (12.10) and (12.12).
Θ	$(m \ H_p) \times (l \ H_u)$	Equal to $\mathcal{C}\mathcal{B}_{\Delta u}$, defined in (12.10) and (12.12).
\mathcal{T}	$(m \ H_p) \times 1$	The reference trajectory over H_p .
\mathcal{E}	$(m \ H_p) \times 1$	Difference between the \mathcal{T} and a free response.
\mathcal{G}	$(l \ H_u) \times 1$	Part of the optimal solution, defined in (12.19).
\mathcal{H}	$(l \ H_u) \times (l \ H_u)$	Part of the optimal solution, defined in (12.19).

Table 12.1: Table containing sizes of matrices and vectors used in the MPC chapter.

12.2 Model Predictive Control of a Standalone Helicopter

Since MPC is slower than most other control strategies due to the prediction steps, and with the fast dynamics of the "Helicopter with slung load" system, the first implementation of MPC will be on top of another, faster, LQ controller that stabilise the system in an initial hover mode, a diagram is shown in figure 12.2, where the illustration indicates what the MPC loop see as the plant.

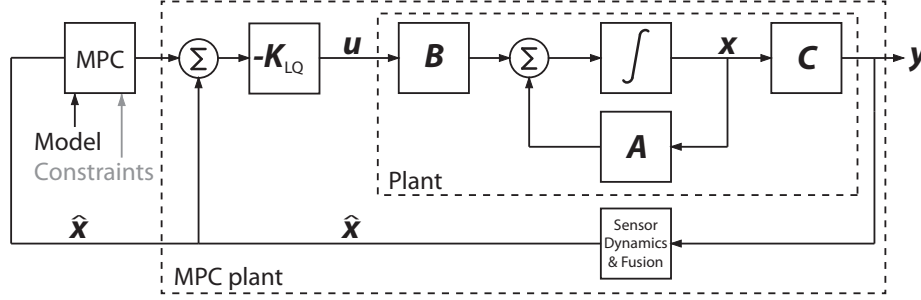


Figure 12.2: A block diagram of the model in state space representation, first with an LQ controller to stabilise the system, with another control loop containing the MPC on top.

In the figure it is assumed that $\mathbf{C} = \mathbf{C}_y = \mathbf{C}_z$. The constraints input to the MPC controller is coloured gray since it is not used in this implementation. It should further be noted that with this kind of implementation with one controller above the other, the controller design should be treated with caution, and make sure that the outer controller operates slower than the inner loop. The LQ controller should be ≥ 10 faster, and have a sufficient bandwidth to follow the reference signals generated by the MPC[9].

Since many of the designed matrices such as the weight matrices grow large with longer horizons, they are not presented in the report, but the design process is documented. The implementation in Simulink is done using the *Model Predictive Control Toolbox* for MATLAB.

12.2.1 Deriving MPC Dependencies

The outputs from the model predictive controller is not the actuator values like with the LQ controller, but rather a set of set points for the input to the LQ controller, meaning a MPC output for each input to the LQ controller. The system seen from the MPC should be as defined in (12.22) using the expanded feedback diagram shown in figure 12.3, with the simplifications in (12.22) assuming that the \mathbf{C} matrix is an identity matrix with the same dimensions as \mathbf{A} .

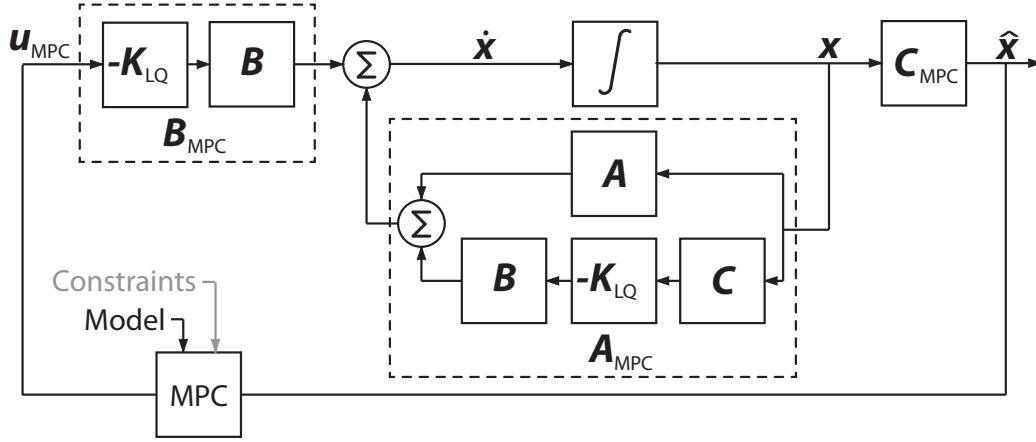


Figure 12.3: An expanded block diagram of the model seen by the model predictive controller, in state space representation.

$$\begin{aligned}
 \mathbf{A}_{MPC} &= \mathbf{A} - \mathbf{B} \mathbf{K}_{LQ} \\
 \mathbf{B}_{MPC} &= -\mathbf{B} \mathbf{K}_{LQ} \\
 \mathbf{C}_{MPC} &= \mathbf{I}_{12 \times 12}
 \end{aligned} \tag{12.22}$$

12.2.2 MPC and LQ Control of a Standalone Helicopter in Hover

The initial set points for the MPC should be the same as for the LQ controller, which is a set of desired operating points where some of those operating points are defined from the trim process, as shown in vector form in (12.23).

$$\mathbf{s}(k)_{set-point, hover} = [0 \ 0 \ -6 \ 0.0516 \ -0.0011 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \tag{12.23}$$

The design of the MPC in MATLAB is handled by `mpctool`, which provides a graphical user interface (GUI) for the different controller values, given a linearised model of the system. The setup of `mpctool` is shown in figure 12.4.

Additional input set in `mpctool` in (12.24) is the prediction and control horizon and the sample time of the predictive controller.

$$Ts = 0.2 \quad H_p = 10 \quad H_u = 2 \tag{12.24}$$

The weights of the LQ controller used to stabilise the inner loop of the system is presented in (12.25). The only state that has a high weight is the one on the z-position of the helicopter.

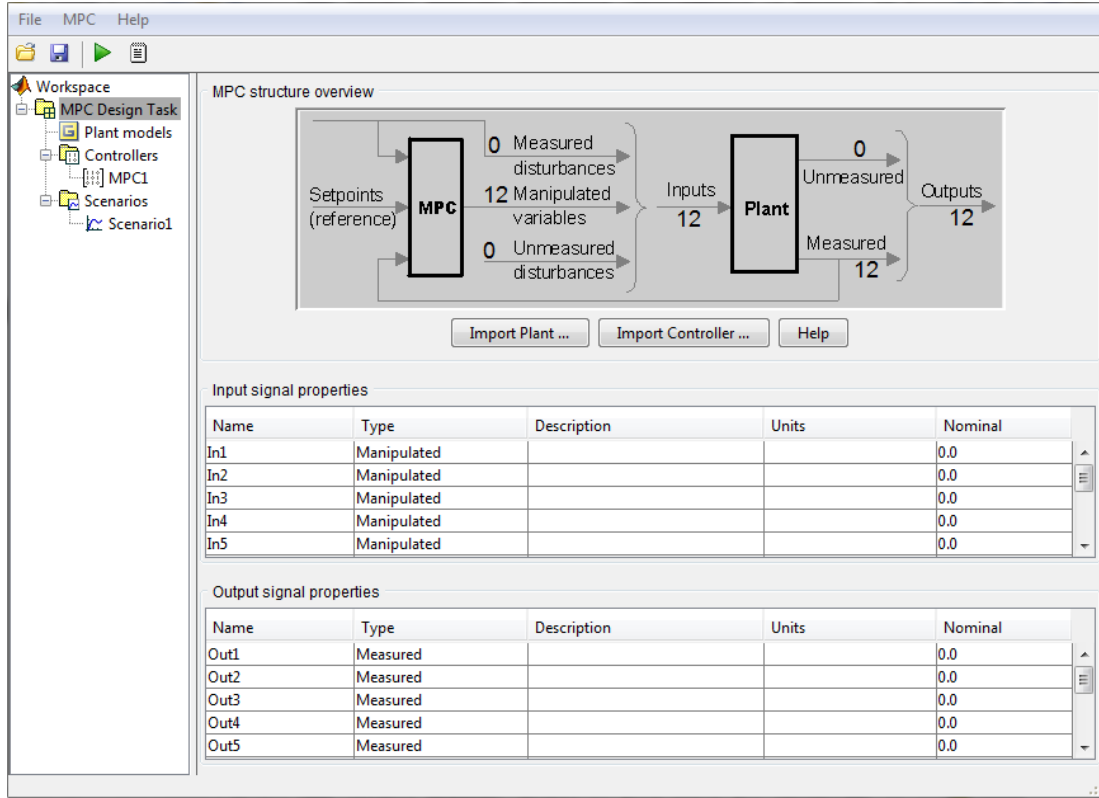


Figure 12.4: A screenshot showing the mpctool GUI.

$$\begin{aligned} \mathbf{Q}_{LQ,diagonal} &= [0.1 \ 0.1 \ 4 \ 0.1 \ 0.1 \ 0.1 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01 \ 0.01] \\ \mathbf{R}_{LQ,diagonal} &= [50 \ 100 \ 100 \ 50] \end{aligned} \quad (12.25)$$

To find out if the inner loop LQ controller, stabilise the system plant for the outer MPC loop, a pole analysis for is performed. The poles is found through the eigenvalues of the \mathbf{A}_{MPC} matrix in (12.22), and presented in table 12.2.

The closed loop poles in the table show that the LQ controller has stabilised the system significantly compared to the open loop system, and is further used to implement the model predictive hover controller.

mpctool implement the reference trajectory $\mathbf{r}(k)$ through change rate weights, meaning how fast each controller output should be allowed to change, defined in \mathbf{S} together with the diagonal entries in \mathbf{Q} and \mathbf{R} in (12.26).

$$\begin{aligned} \mathbf{Q}_{diagonal} &= [5 \ 10 \ 3 \ 1 \ 1 \ 1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1] \\ \mathbf{R}_{diagonal} &= [5 \ 5 \ 10 \ 50 \ 50 \ 50 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000] \\ \mathbf{S}_{rate,weight} &= [2 \ 2 \ 3 \ 5 \ 5 \ 5 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000 \ 1000] \end{aligned} \quad (12.26)$$

LQ controlled hover			
Description	Pole	ζ	ω_n
Single pole	-49.39	1	49.39
Single pole	-7.27	1	7.27
Pole pair	$-4.96 \pm 4.62i$	0.73	6.78
Single pole	-4.42	1	4.42
Pole pair	$-1.34 \pm 1.90i$	0.58	2.32
Pole pair	$-1.10 \pm 1.83i$	0.51	2.13
Pole pair	$-2.06 \pm 0.07i$	0.10	2.07
Single pole	-0.04	1	0.04

Table 12.2: List of poles and their corresponding natural frequency and damping ratio.

Figure 12.5 show the x- and y-reference signals from the MPC given to the inner loop LQ controller, the remaining reference signals are all so small that they negligible. Figure 12.6 show the output signals from the inner loop LQ controller.

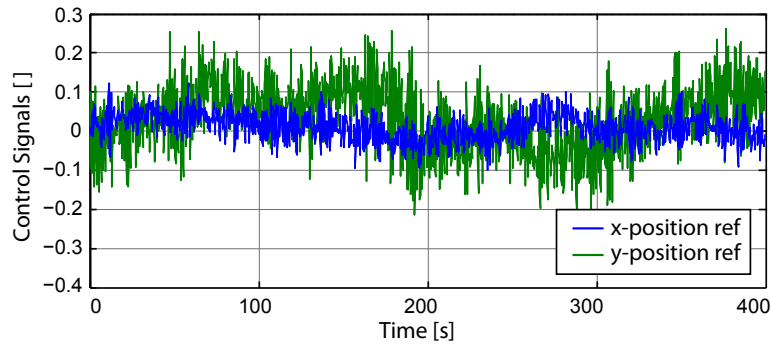


Figure 12.5: Control signals for x- and y- reference outputs from the MPC, for a period of 400 seconds.

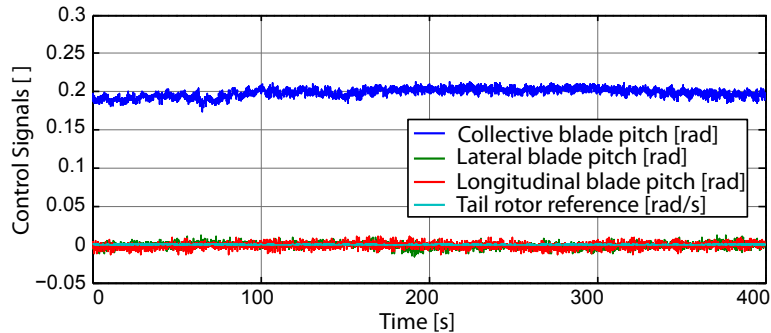


Figure 12.6: Control signals from the inner loop LQ controller, for a period of 400 seconds.

The x-, y- and z-position of the helicopter is shown in figure 12.7 where it is shown that the helicopter is kept close to the reference position at the y- and z-position, but with an offset in the x-position. This offset could be explained from the estimated sensor values, which is investigated further in the first acceptance test in 13.1 on page 142.

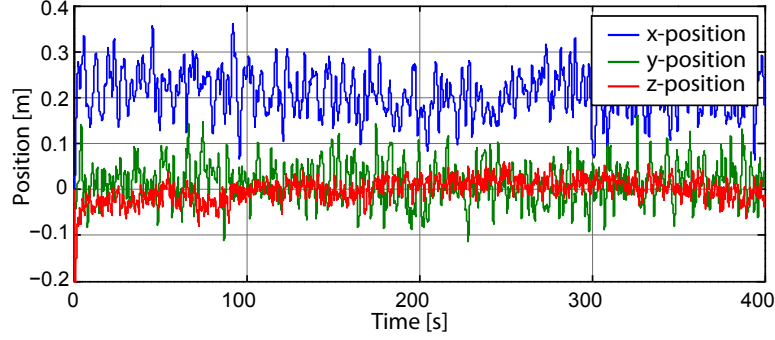


Figure 12.7: x-, y- and z-position of the helicopter for a period of 400 seconds.

12.2.3 MPC and LQ control of a Standalone Helicopter in Flight

The design process for the MPC with the helicopter in flight is similar to the hover controller, where a set of set-points is given to the MPC as an input to which it should generate set-points for the inner loop LQ controller. The set points for forward flight at 6m/s, without any yaw-offset, is as shown in (12.27). The sample time, prediction- and control horizon is as shown in (12.28), both horizons has been increased from the hover controller, to achieve better results.

$$\mathbf{s}(k)_{set-point,flight} = [0 \ 0 \ -6 \ 0.0415 \ -0.0035 \ 0 \ 6 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (12.27)$$

$$Ts = 0.2 \quad H_p = 20 \quad H_u = 5 \quad (12.28)$$

The inner loop LQ controller should in this case have 0 weights on the position, and control on the x- and y-velocity of the controller, as well as the yaw-angle of the helicopter. The weights are shown in (12.29).

$$\begin{aligned} \mathbf{Q}_{LQ,diagonal} &= [0 \ 0 \ 10 \ 0.1 \ 0.1 \ 10 \ 1 \ 0.5 \ 0.1 \ 0.01 \ 0.01 \ 0.01] \\ \mathbf{R}_{LQ,diagonal} &= [40 \ 800 \ 800 \ 1] \end{aligned} \quad (12.29)$$

The closed loop dynamics of the LQ controller generated with the poles shown in (12.29) is examined through the closed loop poles like done with the hover controller. The poles of the closed loop system for forward flight is shown in table 12.3.

The table show the flight system to be stable, with the exception of the second order pole in 0 which is from the xy-position which is not used in the flight controller. With the LQ controller generated from the weights shown in (12.29), it is now possible to

LQ controlled flight			
Description	Pole	ζ	ω_n
Second order pole	0	-	0
Single pole	-48.19	1	48.19
Pole pair	$-8.05 \pm 5.15i$	0.84	9.55
Pole pair	$-2.39 \pm 3.06i$	0.62	3.88
Pole pair	$-2.74 \pm 2.20i$	0.78	3.51
Pole pair	$-4.64 \pm 0.32i$	0.99	4.66
Single pole	-2.25	1	2.25

Table 12.3: List of poles and their corresponding natural frequency and damping ratio.

design an outer loop MPC to generate set-points and reference trajectories for the LQ controller. The weights and rate weights are shown in (12.30).

$$\begin{aligned}
 \mathbf{Q}_{diagonal} &= [1000 \ 1000 \ 30 \ 300 \ 300 \ 15 \ 2 \ 3 \ 100 \ 1000 \ 1000 \ 1000] \\
 \mathbf{R}_{diagonal} &= [0 \ 0 \ 12 \ 0 \ 0 \ 120 \ 30 \ 20 \ 0 \ 0 \ 0 \ 0] \\
 \mathbf{S}_{rate,weight} &= [1000 \ 1000 \ 30 \ 100 \ 100 \ 50 \ 20 \ 25 \ 100 \ 1000 \ 1000 \ 1000] \quad (12.30)
 \end{aligned}$$

Testing the performance of the controller is done by observing the x-velocity of the helicopter, which is given a set-point of 6.0m/s, with the results shown in figure 12.8. It is shown that the controller is able to reach and hold the set-point velocity.

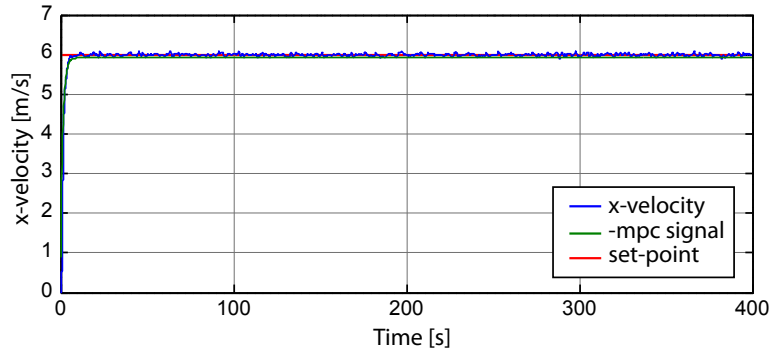


Figure 12.8: x-velocity, mpc reference and set-point for a period of 400 seconds.

While figure 12.8 only show how the controller is able to keep the x-velocity close to the reference, figure 12.9 show the startup, where one of the advantages of MPC is shown. When given a set-point of 6.0m/s to the x-velocity, a reference trajectory is calculated, here as an exponential function, and furthermore, the optimal control signals to reach that reference trajectory over the control horizon is calculated.

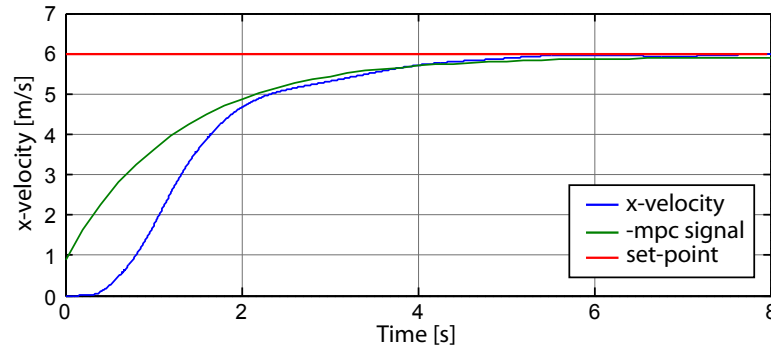


Figure 12.9: x-velocity, mpc reference and set-point for a period of 8 seconds.

It is clear from the startup figure that the inner loop LQ controller is not able to start the helicopter instantaneous, but does catch up to the reference trajectory and is able to follow it closely. The reference trajectory from the MPC is shown negative in the figure, since it is an offset in operating points to the LQ controller. Figure 12.10 show the control signals given to the helicopter by the inner loop LQ controller. The tail-rotor yaw control signal is shown to fluctuate a lot.

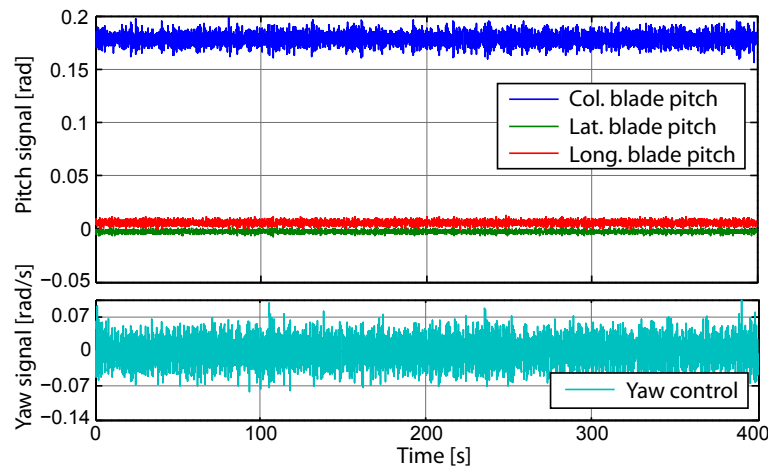


Figure 12.10: Control signals from the inner loop LQ controller for a period of 400 seconds.

The other part of the flight controller is its ability to control the yaw angle of the helicopter, so it is able to fly towards a reference position. When given a yaw reference, a steady state error is observed, so to make up for this offset, an integral state is introduced on the yaw state, to remove this error. The integral state is only included in the outer loop with the MPC, and is thus not part of the inner loop. To prevent windup in the integral state while the helicopter is in hover, the state is reset when a change in reference position is detected.

Figure 12.11 and 12.12 show two tests of yaw angle references set to respectively 1.0 rad and -2.0 rad. The initial rise of the yaw angle reference from the MPC is caused by the normal angle weight, and the slower integral state eventually pull the MPC output up to the set-point. It is chosen to have the integral state work as slow as it does to have as little overshoot as possible, while still being within an acceptable time range.

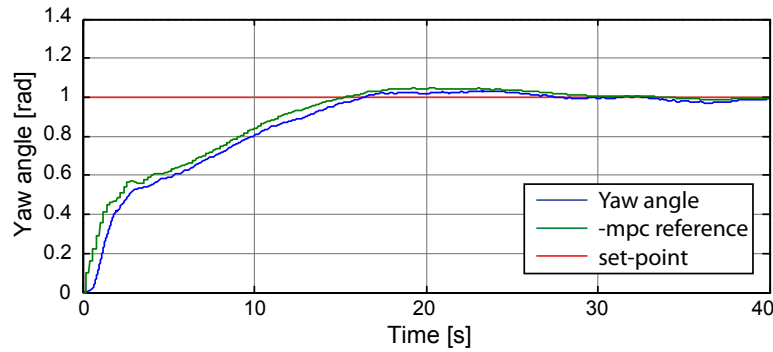


Figure 12.11: Yaw angle, mpc reference and set-point for a period of 40 seconds.

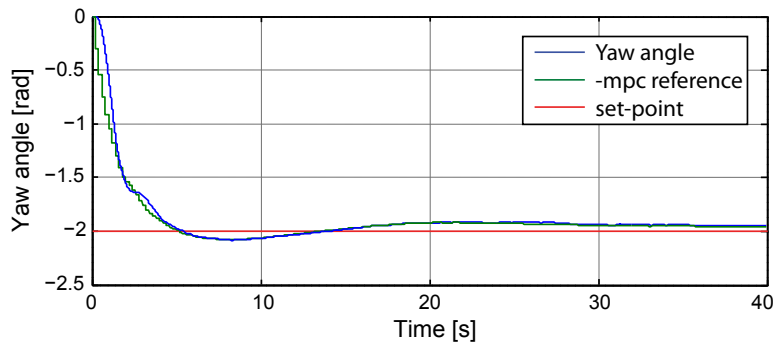


Figure 12.12: Yaw angle, mpc reference and set-point for a period of 40 seconds.

The figures also show that the inner loop is able to follow the reference yaw from the MPC closely, which means that it is the model predictive controller that needs to be faster to reach the set point, if that is wanted.

With the model predictive controllers operating within the ranges of what is needed in this work, the controllers found in this chapter is to be used in the final acceptance test for the standalone helicopter in both flight and hover, with the controllers specified in this section, being an inner loop LQ controller, and an outer loop MPC controlling the references of the inner loop. The state space representation of the inner loop, the MPC and the design of those, as well as the LQ control matrices can be found on the DVD:



```
/Controller/MPC/SAH/SAH_MPC.mat  
/Controller/MPC/SAH/mpc_sah.m  
/Controller/MPC/SAH/MPC_design_hover.mat  
/Controller/MPC/SAH/MPC_design_flight.mat
```

12.3 Model Predictive Control of a Helicopter with Slung Load

With the same requirements to the cascade setup as with the standalone helicopter being in effect for the helicopter with slung load, most of the information needed to make a MPC have already been presented.

The differences between the standalone helicopter test and with a load attached, is the additional states and their dynamics, together with their changes to the already existing dynamics. The linearised system analysis in General Control Considerations show how much the system dynamics changes when a slung load is attached, so the controllers found for the standalone helicopter can not be adopted directly.

The controllers derived in this section includes a MPC for hover, implemented as a set-point controller for an inner loop LQ controller, and a MPC for flight implemented the same way.

12.3.1 MPC and LQ control of a Helicopter with Slung Load in Hover

While the linearised reduced state model for the standalone helicopter had 12 states, the linearised model for the helicopter with load will have 16, with the 4 new ones being the Euler angles describing the position of the slung load, and their angular velocities, updating the state space representation to have the sizes shown in (12.31).

$$\begin{aligned} \mathbf{A}_{MPC,16 \times 16} &= \mathbf{A}_{16 \times 16} - \mathbf{B}_{16 \times 4} \mathbf{K}_{LQ,4 \times 16} \\ \mathbf{B}_{MPC,16 \times 16} &= -\mathbf{B}_{16 \times 4} \mathbf{K}_{LQ,4 \times 16} \\ \mathbf{C}_{MPC,16 \times 16} &= \mathbf{I}_{16 \times 16} \end{aligned} \quad (12.31)$$

In hover, the Euler angles of the load is wanted close to 0, the set-points are equal to the trimmed state values shown in (12.32). And the additional inputs needed before an actual controller design are as shown in (12.33), with the sample frequency and horizons being the same as for the standalone helicopter.

$$\mathbf{s}(k)_{set-point,hover} = [0 \ 0 \ -3 \ 0.05 \ 0.001 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (12.32)$$

$$Ts = 0.2 \quad H_p = 10 \quad H_u = 2 \quad (12.33)$$

The LQ controller in the inner loop should not be as aggressive as for the standalone helicopter, since quick movements is converted directly to the slung load which is not wanted. With that in mind while designing the LQ controller for hover, produced the weightings shown in (12.34), which lower weightings in the \mathbf{Q}_{LQ} matrix, and higher weightings in the \mathbf{Q}_{LQ} matrix.

$$\begin{aligned}
\mathbf{Q}_{LQ,diagonal} &= [1 \ 1 \ 0.5 \ 0.4 \ 0.4 \ 1 \ 1 \ 1 \ 0.1 \ 0.1 \ 0.5 \ 0.1 \ 0.1 \ 0.1 \ 2 \ 2] \\
\mathbf{R}_{LQ,diagonal} &= [160 \ 600 \ 600 \ 5]
\end{aligned} \tag{12.34}$$

Since the inner loop is used to stabilise the system, this is examined through the closed loop poles of the LQ control loop. The closed loop poles are shown in table 12.4.

LQ controlled hover			
Description	Pole	ζ	ω_n
Single pole	-53.20	1	53.20
Single pole	-8.77	1	8.77
Single pole	-6.67	1	6.67
Single pole	-4.80	1	4.80
Pole pair	-2.65±1.89i	0.81	3.25
Pole pair	-1.99±2.18i	0.67	2.95
Pole pair	-1.46±0.45i	0.96	1.53
Single pole	-1.03	1	1.03
Single pole	-1.02	1	1.02
Pole pair	-0.01±0.67i	0.01	0.67
Pole pair	-0.01±0.69i	0.01	0.69

Table 12.4: List of poles and their corresponding natural frequency and damping ratio.

The controller is shown to stabilise the system, but with the last pole pairs being marginally stable, the LQ controller is however still used for further control, with a MPC to control set-points for the inner loop now is needed. The weights used for the hover MPC is shown in (12.35).

$$\begin{aligned}
\mathbf{Q}_{diagonal} &= [10 \ 10 \ 3 \ 1 \ 1 \ 1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1] \\
\mathbf{R}_{diagonal} &= [2 \ 2 \ 25 \ 50 \ 50 \ 50 \ 37 \ 37 \ 1000 \ 1000 \ 1000 \ 10 \ 10 \ 1000 \ 25 \ 25] \\
\mathbf{S}_{rate.weight} &= [8 \ 8 \ 25 \ 5 \ 5 \ 5 \ 25 \ 25 \ 1000 \ 1000 \ 1000 \ 20 \ 20 \ 1000 \ 1 \ 1]
\end{aligned} \tag{12.35}$$

The performance of the produced MPC is first shown through the position of the helicopter which can be seen in figure 12.13, where the deviations from set-points is shown, meaning the z-component has a 3 m offset.

The plot show that the controller is able to keep the helicopter close to the reference position. The deviations from the set-point can partly be explained by the wind acting on the helicopter, so it is interesting to examine what the MPC does for the position offset generated by the wind. The x- and y-position set-points generated by the MPC is shown in figure 12.14.

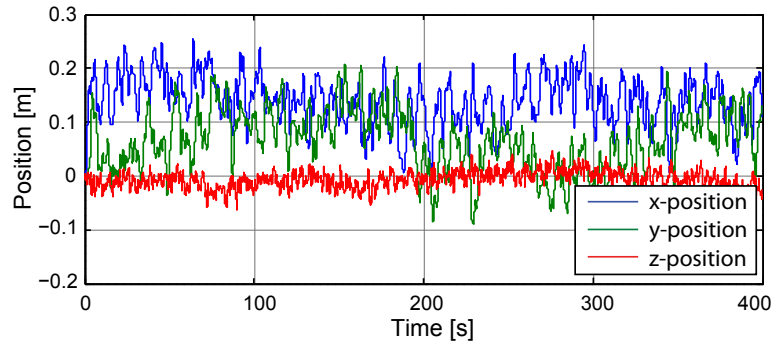


Figure 12.13: Position components of the helicopter, for a period of 400 seconds.

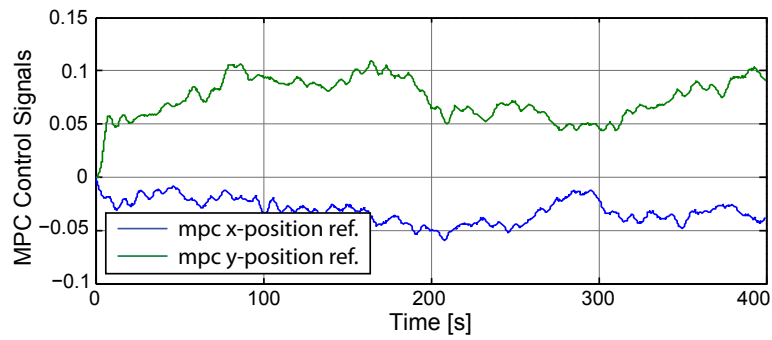


Figure 12.14: MPC Control signals to the LQ controller for x- and y-position of the helicopter, for a period of 400 seconds.

The x- and y-position set-points from the MPC to the LQ controller behave as expected, and generates an offset of the size that is needed to get back to the reference position, additional state offset could be caused by the controller using estimated state values for control, and not true state feedback.

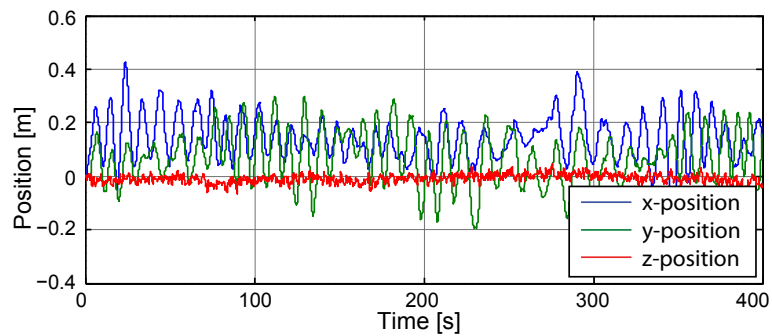


Figure 12.15: Position components of the slung load, for a period of 400 seconds.

Figure 12.15 show the position components of the slung load, where it is shown that the slung load follow the position offset from the reference position somewhat closely, but with larger fluctuations and much slower dynamics, which is expected from the slung load while floating.

The MPC behave as expected, and within acceptable limits, and is further used in model predictive controller section of the acceptance test. The other MPC for the helicopter with slung load system, being the flight controller is found in the following subsection.

12.3.2 MPC and LQ control of a Helicopter with Slung Load in Flight

The MPC for flight with the helicopter and slung load system, does not have to fly as fast as the one for the standalone helicopter, but rather focus on keeping the slung load steady in the water while dragging it. The initial set-points for the MPC is shown in (12.36) together with the sample length as well as the prediction and control horizon are shown in (12.37), where the same considerations as of why to make the horizons longer than in hover remain the same as in the standalone helicopter section.

The integral state that is introduced in the flight controller for the standalone helicopter is adopted to this flight controller for the system with slung load to remove the steady state error from the yaw-angle. This additional state adds a set-point for the MPC input.

$$\mathbf{s}(k)_{set-point,flight} = [0 \ 0 \ -2.85 \ 0.0516 \ -0.0011 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \quad (12.36)$$

$$Ts = 0.2 \quad H_p = 25 \quad H_u = 7 \quad (12.37)$$

The weightings for the inner loop LQ controller is choosen to focus heavily on the height of the helicopter, since that is shown to have great effects on the behaviour of the buoy in the LQ control section 11.3 on page 97, as well as the yaw angle for direction and x- and y-velocities of the helicopter for steady flight. And with very heavy restrictions on the control signals, to reduce quick movement from the helicopter. The weightings are shown in (12.38).

$$\begin{aligned} \mathbf{Q}_{LQ,diagonal} &= [0 \ 0 \ 30 \ 1 \ 1 \ 3 \ 1 \ 1 \ 5 \ 5 \ 2 \ 0.1 \ 0.1 \ 0.1 \ 2 \ 2] \\ \mathbf{R}_{LQ,diagonal} &= [300 \ 1200 \ 1200 \ 40] \end{aligned} \quad (12.38)$$

A stability check for the LQ control loop is needed, and is done by checking the poles of the closed loop system with the inner loop LQ controller. The poles are shown in table 12.5.

The inner closed loop system with the LQ controller is shown to stabilise the system at the operation points, and is thus used in the remaining part of this MPC chapter

LQ controlled flight			
Description	Pole	ζ	ω_n
Second order pole	0	-	0
Single pole	-48.10	1	48.10
Single pole	-8.85	1	8.85
Single pole	-6.68	1	6.68
Single pole	-4.51	1	4.51
Single pole	-3.49	1	3.49
Pole pair	-1.83±1.88i	0.70	2.62
Pole pair	-1.45±1.73i	0.64	2.26
Pole pair	-0.07±1.05i	0.07	1.06
Pole pair	-1.06±0.11i	0.99	1.07
Single pole	-0.36	1	0.36

Table 12.5: List of poles and their corresponding natural frequency and damping ratio.

for the flight controller of the helicopter with slung load system.

Weightings for the flight MPC are shown in (12.39),

$$\begin{aligned}
\mathbf{Q}_{diagonal} &= [1000 \ 1000 \ 1000 \ 300 \ 300 \ 30 \ 250 \ 250 \ 15 \ 17 \ 100 \ 1000 \ 1000 \ 1000 \ 50 \ 50] \\
\mathbf{R}_{diagonal} &= [0 \ 0 \ 12 \ 0 \ 0 \ 120 \ 2 \ 2 \ 60 \ 60 \ 5 \ 0 \ 0 \ 0 \ 5 \ 5 \ 50] \\
\mathbf{S}_{rate,weight} &= [1000 \ 1000 \ 1000 \ 100 \ 100 \ 100 \ 120 \ 500 \ 500 \ 50 \ 57 \ 1000 \ 1000 \ 1000 \ 1000 \ 60 \ 50]
\end{aligned} \tag{12.39}$$

The first performance check for the MPC flight controller is forward flight while dragging the load. Figure 12.16 show the x-velocity of the helicopter together with the negative reference trajectory and set-point, together with the startup when going from hover to flight.

The velocity plot show that how the x-velocity reaches the set-point and is controlled around that point, with the startup plot showing how the x-velocity has overshoot while starting, and does not follow the reference trajectory very closely. The final velocity is however within acceptable values based on the plot.

The helicopter does not only have to start moving itself when going from hover to flight, but now also accelerate the slung load floating in the water. That could explain the overshoot and oscillations on the x-velocity startup plot, so figure 12.17 show the x-velocity of the helicopter again, and the slung load x-velocity so they can be compared.

The two figure show how the x-velocity of the slung load fluctuates more than that of the helicopter, which is also shown in second Euler angle, that oscillates. These results are however what is to be expected, and is acceptable as long as the load angle changes are as slow and with small oscillations as shown in figure 12.17.

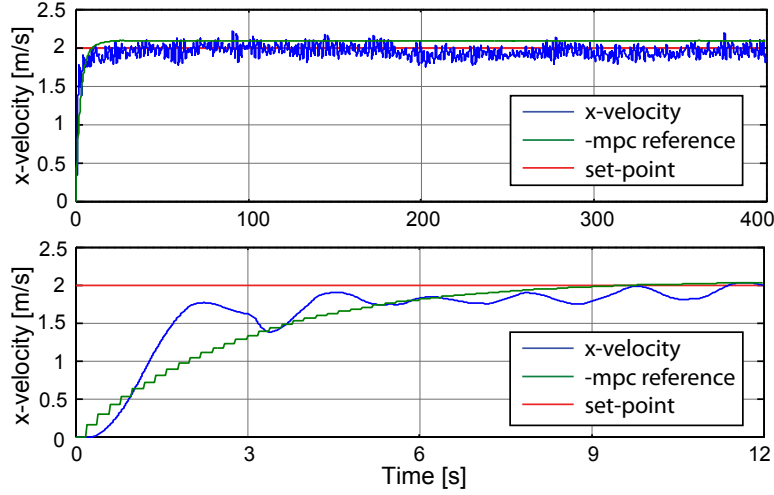


Figure 12.16: x-velocity of the helicopter with negative reference trajectory and set-point, for a period of 400 seconds. And for the 12 second startup.

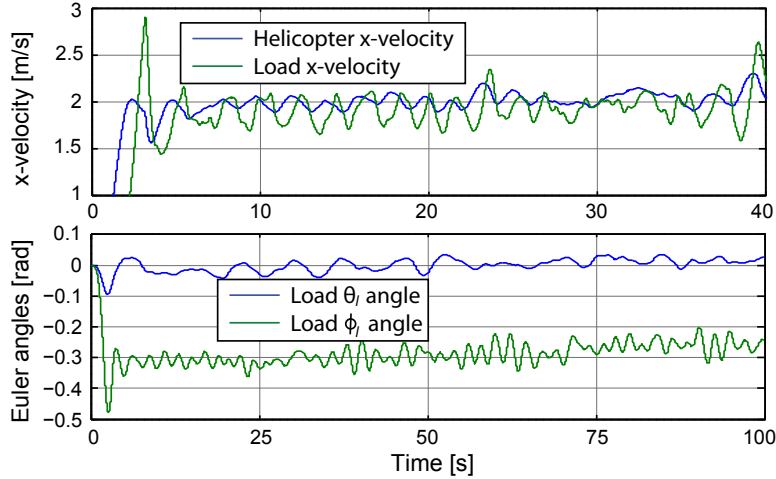


Figure 12.17: x-velocity of the helicopter and slung load with the first and second Euler angles of the slung load for a period of 40 seconds.

While flying forward in the x-direction the controller is able to drag the slung load as specified, but it is yet to be tested if it is able to change yaw-angle, to point towards a reference position. Figure 12.18 show the yaw-angle at a test with 1.0rad and -2.0rad yaw offset.

Both figures show how the yaw angle of the helicopter is changes towards the yaw-reference. The second test did however introduce oscillations to the system, where the slung load is no longer being dragged at an acceptable depth, but went in and out of the water. The x-velocity of the helicopter is shown in figure 12.19 to illustrate the effect on the helicopter.

A velocity set-point rotation is implemented to get better results with tests regarding flight to a reference position, where the x- and y- velocity set-points are rotated with

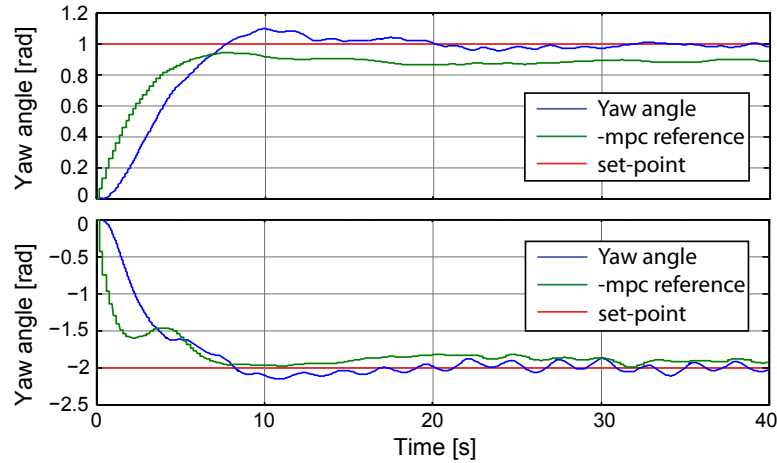


Figure 12.18: Helicopter yaw angle shown with reference trajectory and set-point, for a period of 40 seconds.

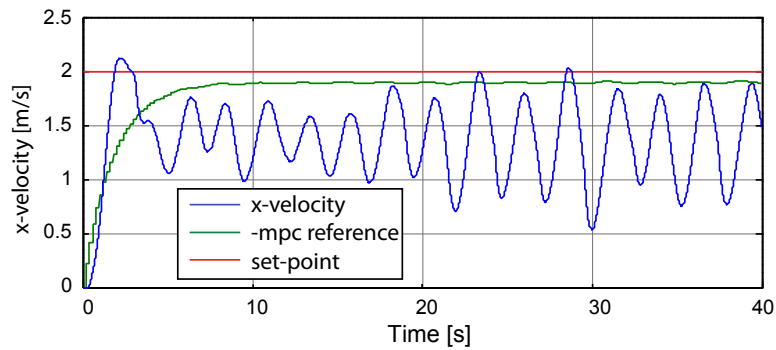


Figure 12.19: Helicopter x-velocity shown with reference trajectory and set-point, for a period of 40 seconds.

the yaw-error which is the angle between the helicopter and the reference position, so the helicopter would start flying towards the reference position, even though the helicopter had not turned yet. Figure 12.20 show the yaw-angle with the velocity correction implemented. and figure 12.21 show the x- and y-velocities of the helicopter for comparison with figure 12.19.

The results are not as good as for forward flight, where the helicopter has the right heading before beginning flight, but it is within acceptable limits, based on the results shown in figures 12.20 and 12.21. The controller presented here is used for further testing in the acceptance test, and the presented material can be found on the DVD:



```
/Controller/MPC/HAL/HAL_MPC.mat
/Controller/MPC/HAL/mpc_hal.m
/Controller/MPC/HAL/MPC_design_load_hover.mat
/Controller/MPC/HAL/MPC_design_load_flight.mat
```

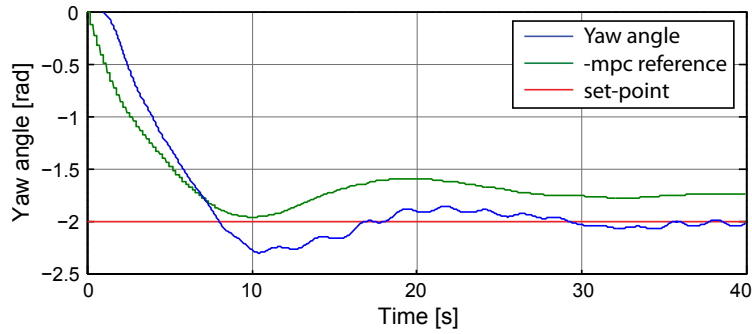


Figure 12.20: Helicopter yaw angle shown with reference trajectory and set-point, for a period of 40 seconds.

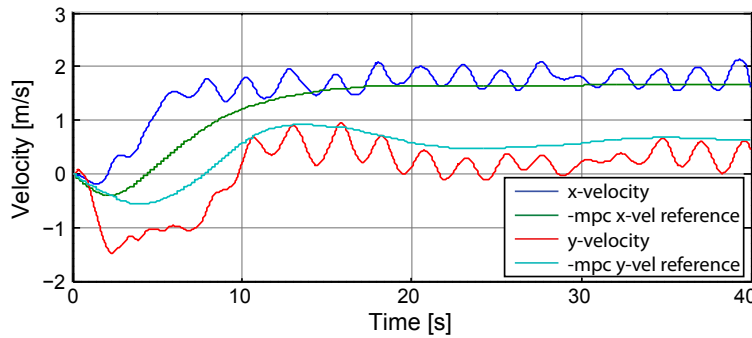


Figure 12.21: Helicopter x- and y-velocity shown with reference trajectories, for a period of 40 seconds.

12.4 Model Predictive Control Conclusion

The model predictive control implemented in this work is the simple version without constraints and disturbance model, leaving the model predictive controller to be the essentials of a linear quadratic controller with a prediction part, calculating the control matrix over the prediction horizon. This leaves the MPC chapter with plenty of expansion opportunities for future work, where the most obvious choice would be to start with the constraints, which is where MPC-LQ setup get the best results.

The controllers that is produced in this chapter included MPC controllers in cascade with an inner loop LQ controller to stabilise the system, here being hover and flight of a helicopter both with and without load. The inner loop LQ controllers are checked to stabilise the system, leaving the MPC to control references for the inner loop, which is done within acceptable limits to further use the controllers in the acceptance test.

Most of the control states for flight and hover is the same for the MPC as with the LQ controller, but to make the MPCs better at controlling the yaw angle of the helicopter, an integral state is implemented on the outer loop, and never seen by the inner loop. With the integral state implemented on the flight controllers, acceptable results for turning are achieved.

Summary of part III

The general control strategies chapter specified the two control strategies used in this work would be Linear Quadratic Control and Model Predictive Control. The operating points and ranges for the control states and input signals was determined, with an appendix added going through and presenting the trimming and linearisation of the helicopter model using Helisim.

A linear model analysis investigated whether control was needed on the system plant which is the helicopter, and found that the uncontrolled system both with and without slung load, would be unstable.

In the LQ control chapter the use of the control strategy was first examined, and afterwards used to find a series of controllers used on both a standalone helicopter and a helicopter with slung load system, to be used in the acceptance tests in the evaluation part of this work.

The second control strategy being Model Predictive Control was implemented as cascade control, with an underlying LQ controller stabilising the system, and then an outer control loop specifying the set-points for the inner control loop. The model predictive controllers produced in this chapter did behave as designed, and is further used in the acceptance test.

Part IV

Evaluation

Part introduction

Contents

13 Acceptance Test

14 Comparison of the Controllers

15 Conclusion

Purpose

The first chapter of the part include the acceptance test of this work, evaluating the performance of the computed controllers in a set of specified tests, where the requirement specifications needs to be fulfilled for the tests to be successful.

A controller comparison chapter is included to test the two control strategies used in this work up against each other, to find the strengths and weaknesses of both when compared.

The conclusion chapter will conclude whether the subparts and the project in a whole have been a success, and is continued into further work that specify some of the areas that needs further work to make the project more successful.

Goals

- *Perform the acceptance tests specified in chapter 4.3*
- *Compare the performance of the MPC-LQ and LQ controller setups.*
- *Conclude on the status and accomplishments of this work.*
- *Specify expansion opportunities for both the modelling and control parts of this work.*

Acceptance Test

This chapter conduct the acceptance test and presents the results. The objectives of the acceptance tests are specified in the acceptance test specification section 4.3 on page 16. The tests are conducted using the Helisim simulation tool in MATLAB and Simulink, with the measured states coming from the onboard state value estimator, which contains sensor fusion and measurement noise. The setup is shown in figure 13.1.

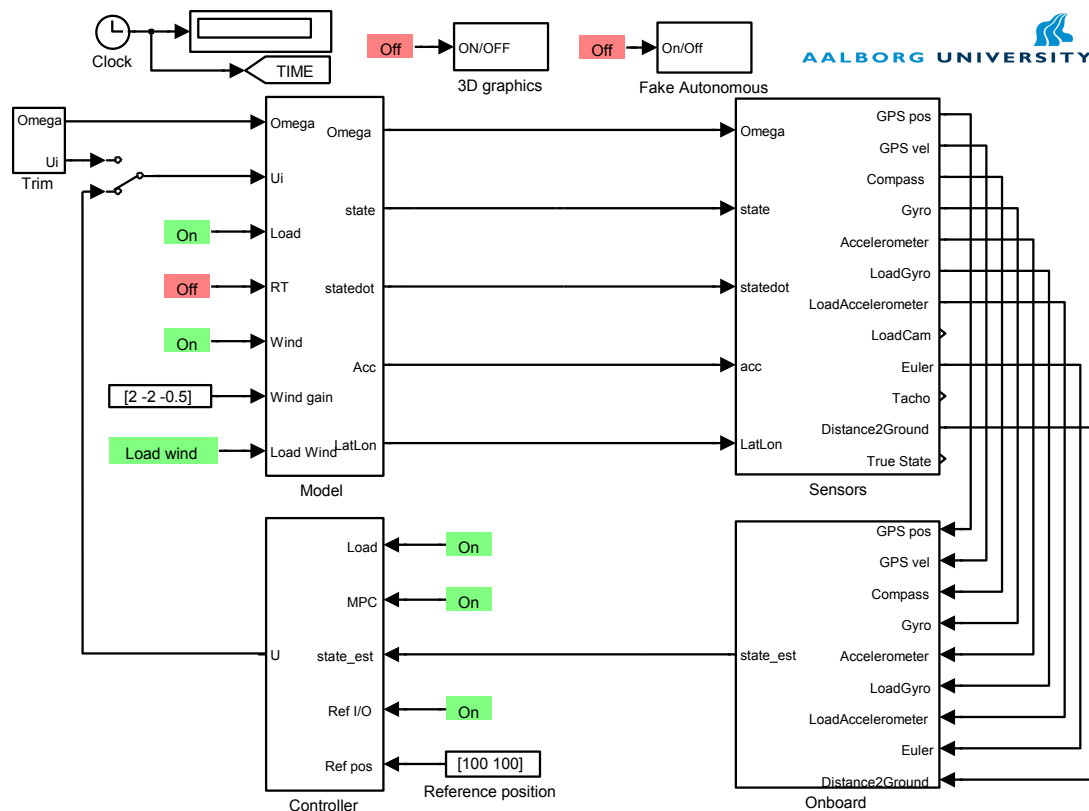


Figure 13.1: Simplified illustration of the Helisim setup used in the acceptance tests.

Apart from the noise on the estimated state values, adding an increased need for robustness in the controllers, wind is also enabled in the acceptance tests. This is done to simulate the controllers performance in a real scenario. All acceptance tests are conducted 10 times, with 10 pregenerated wind inputs, for better comparablilty between tests. The plots shown throughout the acceptance test are generated using the first of the winds, which is shown in figure 13.2.

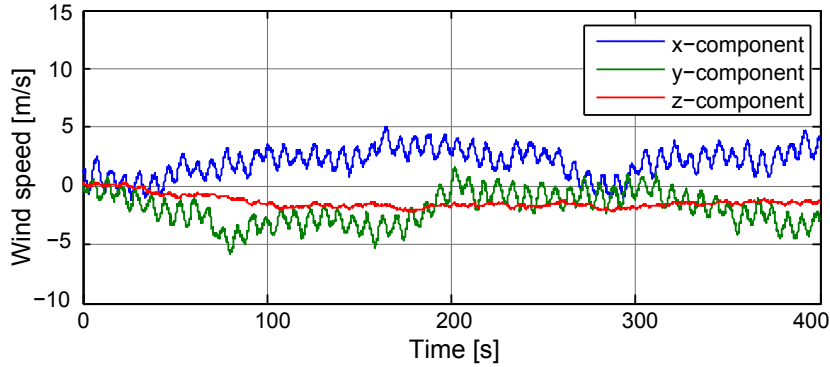


Figure 13.2: A plot of the standard wind used in all of the simulations shown in the acceptance tests, over a period of 400 seconds.

The wind data, along with the estimated and true state values and the control signals can be found on the DVD, with a folder for each acceptance test:



/Acceptance Test/data/...

The following sections will describe the results of the conducted acceptance tests, starting with those for the standalone helicopter, and followed by the ones for the full system requirements. Each requirement is provided its own section, by the name of the requirement, each with a subsection for LQ control and MPC-LQ control.

13.1 Standalone helicopter acceptance tests

HP1: Hovering of the Helicopter, in a Steady Position Above a Certain Point

As stated in the acceptance test specification, section 4.3 on page 16, the helicopter is set to hover at a fixed position for 400 seconds, and should at all time be within 0.50 meters horizontally and 0.50 meters vertically. The horizontal displacement is calculated by the Euclidean norm of the x- and y-position.

The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

Figure 13.3 shows the altitude and horizontal error of the helicopter during the 400 seconds of the simulation.⁷

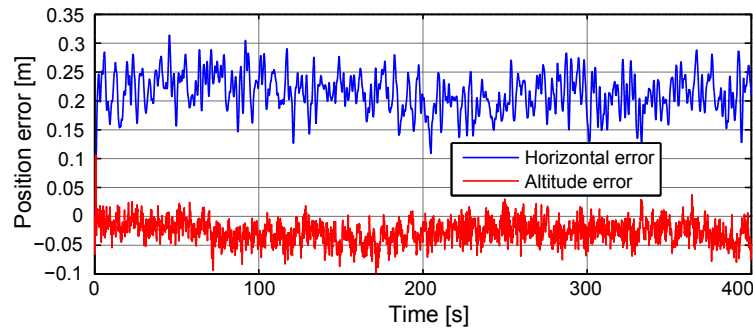


Figure 13.3: This plot shows the altitude and horizontal error of the helicopter during the 400 seconds of simulation.

It can be seen that the helicopter at all times is within the specified requirement of 0.50 meters in horizontal and vertical displacement, and that there appears to be a bias in the horizontal displacement. Of the 10 simulations that is run for this test with only the LQ controller, the maximum error in altitude is 0.14 m and a maximum error in horizontal displacement is 0.37 m (all numbers for requirements are rounded up).

Figure 13.4 shows the errors in the estimated positions, which is what the controller sees, and tries to control to zero.

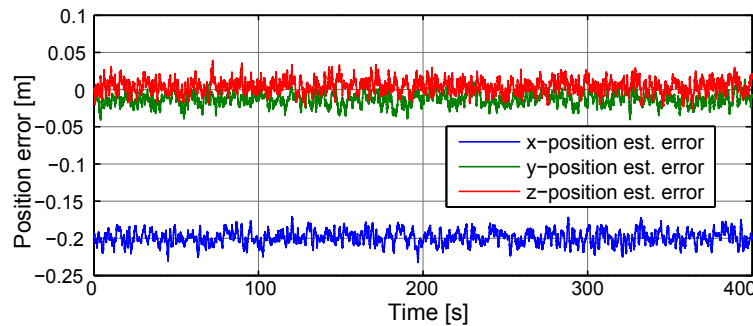


Figure 13.4: The position estimation error of the helicopter during the 400 seconds of simulation.

It appears that there is a 0.2 m bias on the estimated x-position, that is not accounted for. This bias could be a result of the placement of the IMU on the helicopter. Table 13.1 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Horizontal error	0.50 m	0.37 m	✓
Altitude error	0.50 m	0.14 m	✓

Table 13.1: HP1.1 results and verdict.

With MPC and LQ Control

In this setup with an inner loop LQ controller handling the fast dynamics of the system and the wind disturbance, and a MPC in cascade to control the LQ set-points of the LQ controller, this setup should be able to produce results as good or better than with only an LQ controller.

Figure 13.5 show the horizontal and altitude errors from the reference position of (0,0) m at an altitude of 3 m, while the standalone helicopter is in hover.

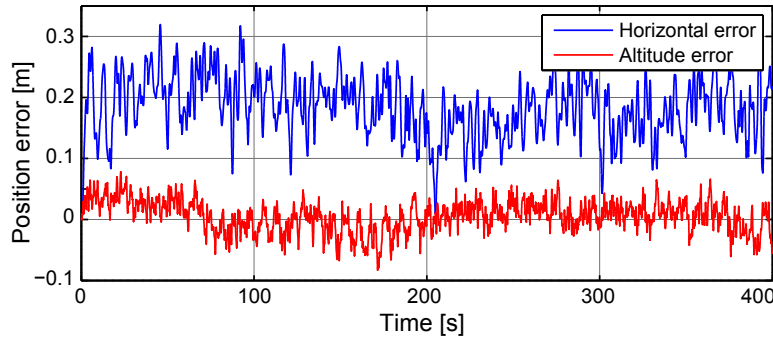


Figure 13.5: This plot shows the altitude and horizontal error of the helicopter during the 400 seconds of simulation.

The figure show how the cascade controller is able to keep both errors within the specified limit of 0.5 m. When comparing the results to those of the LQ controller alone shown in figure 13.3 on the preceding page. The MPC-LQ setup is able to control the position closer to the reference, but with larger fluctuations than with the LQ controller alone.

The error results for the simulation is shown in the figure is a maximum altitude error of 0.09 m and maximum horizontal deviation of 0.31 m, which is well within the requirement limit. The worst case results for the 10 simulations show a maximum altitude deviation of 0.20 m and horizontal error of 0.40 m, which is closer to the requirement limit, but still passed. The results, requirements and verdict are shown in table 13.2.

	Requirement	Result	Verdict
Horizontal error	0.50 m	0.40 m	✓
Altitude error	0.50 m	0.20 m	✓

Table 13.2: HP1.2 results and verdict.

The scripts used to generating these plots, as well as the data from the simulations is available on the enclosed DVD, along with additional plots.



/Acceptance Test/data/HP1/...

HP2: Flying Towards a Reference Position

It is specified in the acceptance test specification, section 4.3 on page 16, that the helicopter is required go from hover and fly towards a specified reference position at $\{2500, 2500\}$, with a maximum velocity error of 1.00m/s towards that position. The transition from hover to flight is required to take less than 10.00 seconds and while in transition, it is not allowed to deviate more than 0.50 meters in altitude.

The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

Figure 13.6 shows that the helicopter position error, which verifies that the helicopter does fly towards the specified point.

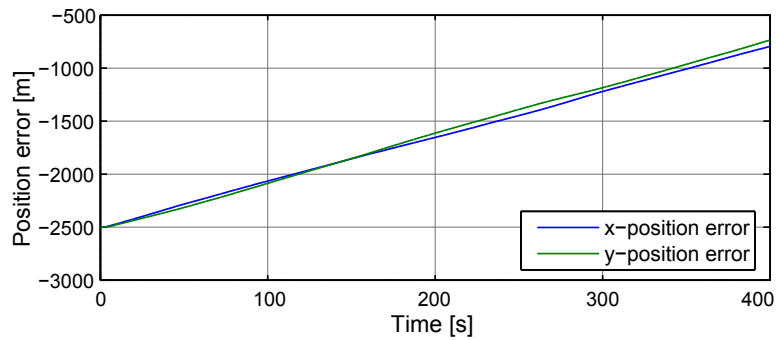


Figure 13.6: This plot shows the position error of the helicopter during the 400 seconds of simulation.

It can be seen that it flies towards the reference position at uneven rates, this is however not always the case in all of the 10 simulation, which indicates that it is caused by the wind.

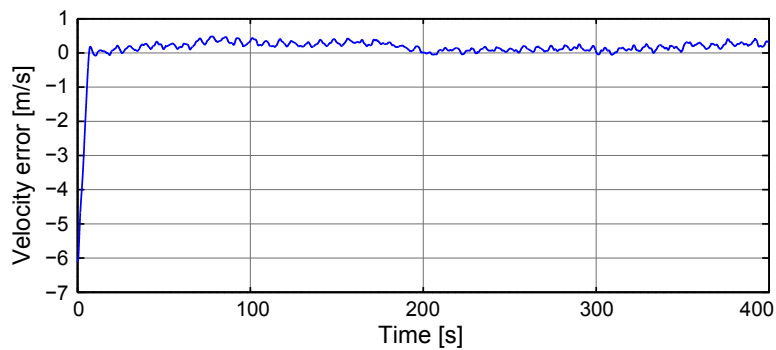


Figure 13.7: This plot shows the velocity error of the helicopter towards the target position during the 400 seconds of simulation.

Figure 13.7 on the preceding page shows that the velocity of the helicopter towards the destination deviates by less than the 1.00m/s limit, which it is 5.78 seconds to reach. During this 5.50 second transition the maximum error in altitude is 0.37 meters, as shown in figure 13.8.

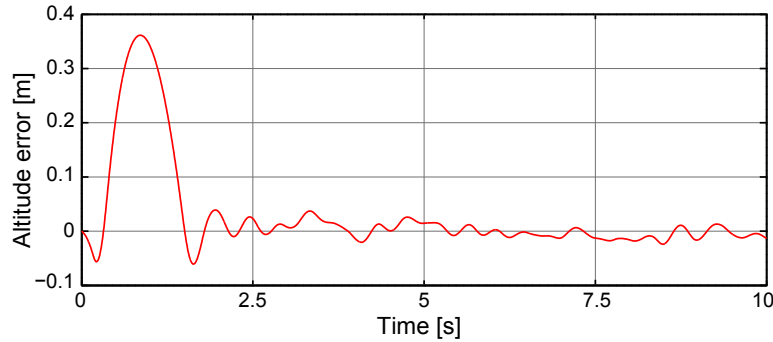


Figure 13.8: This plot shows the altitude of the helicopter during the transition from hover to flight.

In all of the 10 simulations conducted for this requirement, the highest error from the velocity reference is 0.85m/s , which is also below the specified limit, the slowest transition time is 5.78 seconds, and the largest altitude error is 0.37 meters. Table 13.3 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Velocity error	1.00m/s	0.85m/s	✓
Transition time	10.00 s	5.78 s	✓
Altitude error	0.50 m	0.37 m	✓

Table 13.3: HP2.1 results and verdict.

With MPC and LQ Control

The helicopter is given a reference position outside its reach within the simulation, to observe the flight trajectory and the transition from hover to flight. Figure 13.9 on the next page show the position error from the set-point. With the MPC being able to give a reference for side-slip to counteract the wind, the x- and y-positions follow each other closely.

While flying towards the reference position, it is subject to a requirement on the velocity towards the reference position. Figure 13.10 on the facing page show how the velocity is kept around the set-point after the initial rise.

To avoid a low altitude, a requirement of 1.00 m is set on the altitude drop the system is allowed to have. Figure 13.11 on the next page show the drop on the transaction between the two controllers, which is where the largest drop is observed.

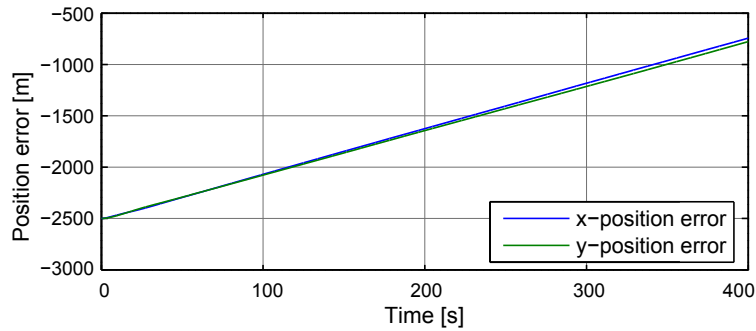


Figure 13.9: This plot shows the position error of the helicopter during the 400 seconds of simulation.

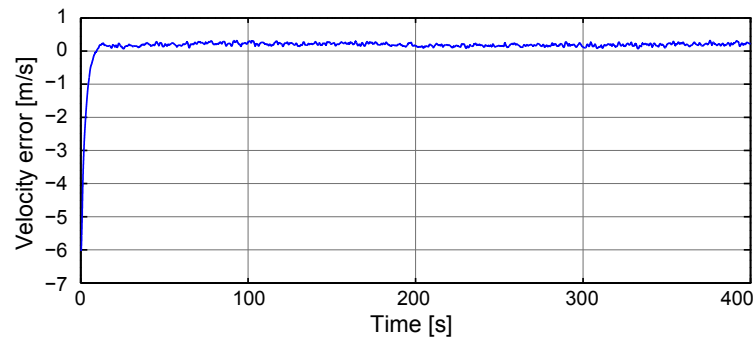


Figure 13.10: This plot shows the velocity error of the helicopter towards the target position during the 400 seconds of simulation.

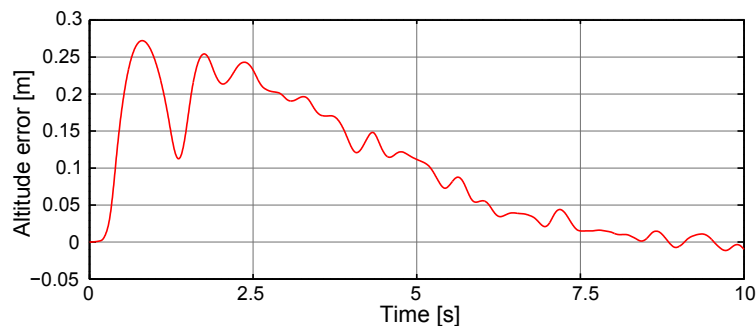


Figure 13.11: This plot shows the altitude of the helicopter during the transition from hover to flight.

Of the 10 simulations run for this test, the worst case results for velocity error is 0.38 m/s , a maximum altitude error of 0.28 m and a worst case transitions time of 4.65 s , which are all better results than with just the LQ controller. The requirement verdict is shown in table 13.4 on the following page.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/HP2/...

	Requirement	Result	Verdict
Velocity error	1.00m/s	0.38m/s	✓
Transition time	10.00 s	4.65 s	✓
Altitude error	0.50 m	0.28 m	✓

Table 13.4: HP2.2 results and verdict.**HS3: Arriving at a Reference Position**

This acceptance test concerns the part of the mission where the helicopter arrives at the location, where it should be able to go from flight to hover mode. As stated in the acceptance test specification, section 4.3 on page 16, the helicopter should be able to make this transition within 10.00 seconds, without deviating more than 0.50 meters in altitude. Once in hover mode, it must not deviate more than 0.50 meters horizontally and vertically.

The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

Figure 13.12 shows the horizontal position errors of the helicopter, while it flies towards the specified location, and ones at the location a close-up is shown where it can be seen that it stays within the 0.50 meter boundary.

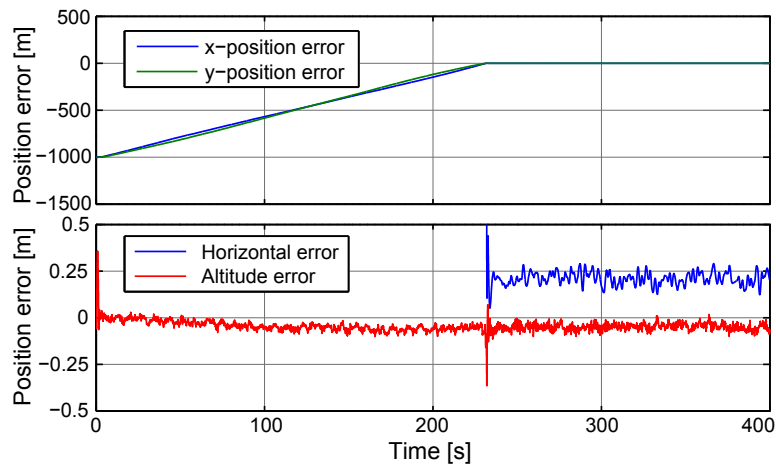


Figure 13.12: This plot shows the horizontal position, and altitude error of the helicopter during the 400 seconds of simulation. The top plot shows the horizontal positions during the entire flight, while the bottom plot shows a close-up on the hover region, where the altitude also is plotted.

In all of the 10 simulations the largest horizontal position error is 0.43 meters and the largest altitude error is 0.45 meters.

Figure 13.13 shows the Horizontal position error and the altitude of the helicopter during the transition phase, where it can be seen that neither of them violates the 0.50 meter limit, set forth by the requirements.

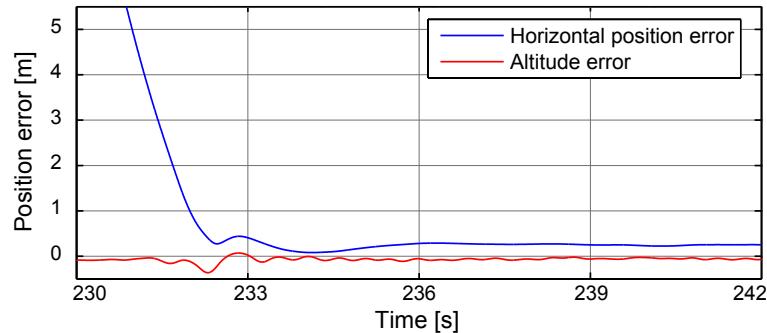


Figure 13.13: This plot shows the horizontal position and altitude error of the helicopter during the transition from flight to hover.

The transition from flight to hover, shown on figure 13.13 takes 1.29 seconds, and has a maximum error in altitude of 0.34 meters. In all of the 10 simulations for this requirement, the slowest transition is 2.97 seconds and the largest altitude error is 0.46 meters.

Table 13.5 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Hover horizontal position error	0.50 m	0.43 m	✓
Hover altitude error	0.50 m	0.45 m	✓
Transition time	10.00 s	2.97 s	✓
Transition altitude error	0.50 m	0.46 m	✓

Table 13.5: HS3.1 results and verdict.

With MPC and LQ Control

Here a reference position that is reached during the simulation is given, to observe the transition from flight to hover when the reference position is reached. Figure 13.14 show the x-, y-position error over the simulation, as well as a closeup of the horizontal and vertical errors, with results very close to those of the LQ controller alone.

Figure 13.15 on the following page show the errors in the period of the flight to hover transaction, again with large similarities to that of the LQ controller, which is evident since the transaction is handled the same way in both cases.

The results of the transition is shown in table 13.6 on the next page with the worst case result being nearly identical to that of the LQ controllers alone. With a worst case horizontal error of 0.47 m, and altitude error of 0.41 m being very close to the maximum

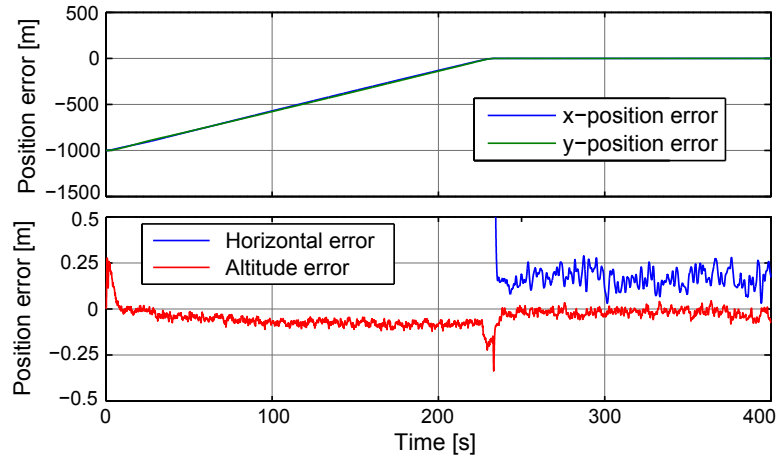


Figure 13.14: This plot shows the horizontal position, and altitude error of the helicopter during the 400 seconds of simulation. The top plot shows the horizontal positions during the entire flight, while the bottom plot shows a close-up on the hover region, where the altitude also is plotted.

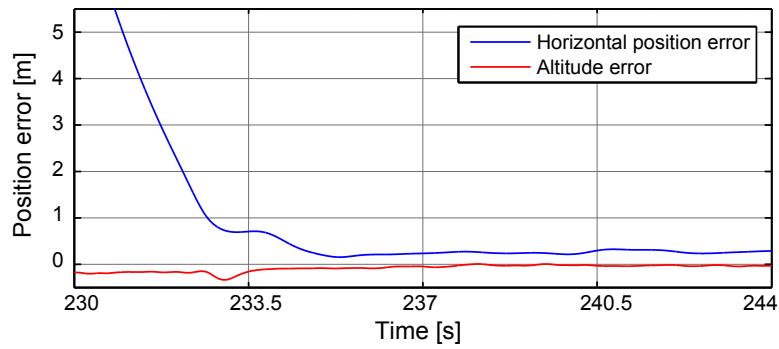


Figure 13.15: This plot shows the horizontal position and altitude error of the helicopter during the transition from flight to hover.

errors allowed by the requirement of 0.50 m, it is clear that the transition from flight to hover could be improved, but seen as this is a secondary requirement, it is not further investigated.

	Requirement	Result	Verdict
Hover horizontal position error	0.50 m	0.47 m	✓
Hover altitude error	0.50 m	0.41 m	✓
Transition time	10.00 s	4.15 s	✓
Transition altitude error	0.50 m	0.43 m	✓

Table 13.6: HS3.2 results and verdict.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/HS3/...

13.2 Full system acceptance tests

SP1: Hovering of the Helicopter, in a Steady Position above the Semi-submerged Load, with a Tense Cable

As stated in the acceptance test specification, section 4.3 on page 16, the helicopter with semi-submerged slung load is set to hover at a fixed position at 3 meters altitude for 400 seconds, and should at all time be within 0.50 meters horizontally and 0.50 meters vertically of that position. The horizontal displacement is calculated by the Euclidean norm of the x- and y-position. The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

The horizontal displacement and the altitude error of the helicopter is shown on figure 13.16, where it can be seen that both are within the 0.50 meter limit, put forth by the requirements.

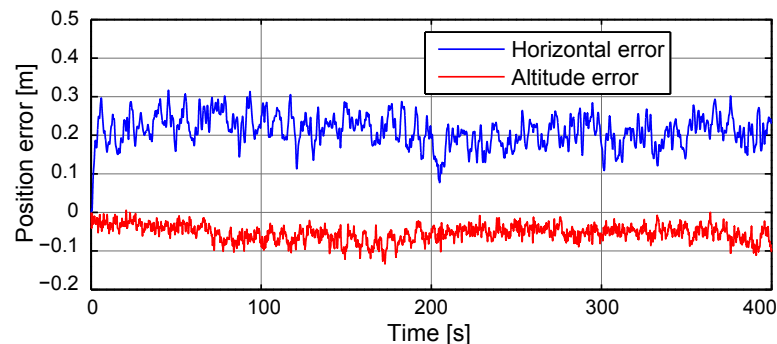


Figure 13.16: This plot shows the altitude and horizontal error of the helicopter during the 400 seconds of simulation.

The maximum errors for the first test is 0.32 and 0.14 meters for the horizontal and altitude error respectively. If all of the 10 simulations are considered, the maximum errors are: 0.37 and 0.29 meters respectively. Table 13.7 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Horizontal position error	0.50 m	0.37 m	✓
Altitude error	0.50 m	0.29 m	✓

Table 13.7: SP1.1 results and verdict.

With MPC and LQ Control

With the slung load attached, the results with a MPC reference controller is not expected to improve much when compared with the LQ controller alone. Figure 13.17 show the position errors from the reference, split up into the values that had requirements on them.

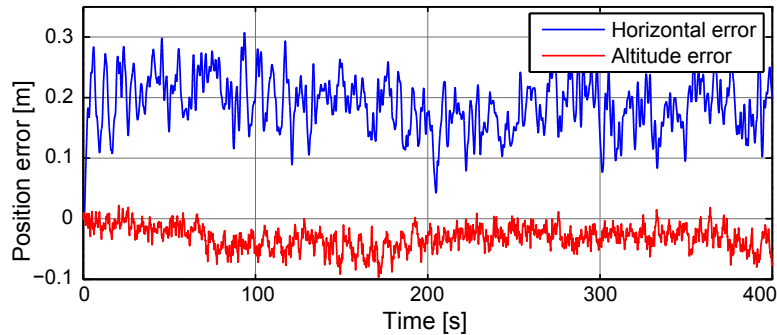


Figure 13.17: This plot shows the altitude and horizontal error of the helicopter during the 400 seconds of simulation.

Both the horizontal and altitude errors have a requirement of 0.50 m maximum allowed error. The worst case results for the 10 simulations for this test revealed results of a horizontal error of 0.37 m and an altitude error of 0.25 m, which is both within the requirements. Table 13.8 show the results and verdict.

	Requirement	Result	Verdict
Horizontal position error	0.50 m	0.37 m	✓
Altitude error	0.50 m	0.25 m	✓

Table 13.8: SP1.2 results and verdict.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/SP1/...

SP2: Flying Towards a Reference Position

This test treats flying towards a specified reference position outside the reach of the helicopter, with semi submerged load attached. The helicopter is started in hover and given a reference it should then start flying towards the specified location, spending no more than 10.00 seconds transcending from hover to flight mode. In the transition it is not allowed to deviate more than 1.00 meter in altitude. Once in flight mode, the velocity towards the target must not deviate by more than 1.00m/s.

The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

Figure 13.18 shows that the helicopter position error, which verifies that the helicopter does fly towards the specified point.

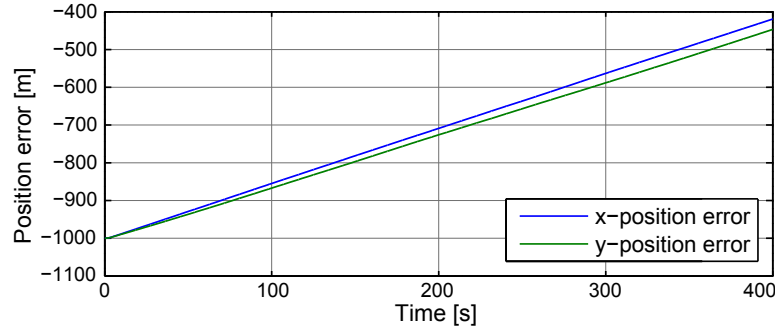


Figure 13.18: This plot shows the position error of the helicopter with load during the 400 seconds of simulation.

It is shown that the x- and y-velocities also here are different, even more that what is shown for the standalone helicopter, this can be blamed on the wind having bigger effects on the helicopter with increased weight, and more unstable dynamics.

Figure 13.20 on the following page shows the helicopters velocity towards the target position. The transition time of this particular test is 1.86 seconds, well within the limit of 10.00 seconds.

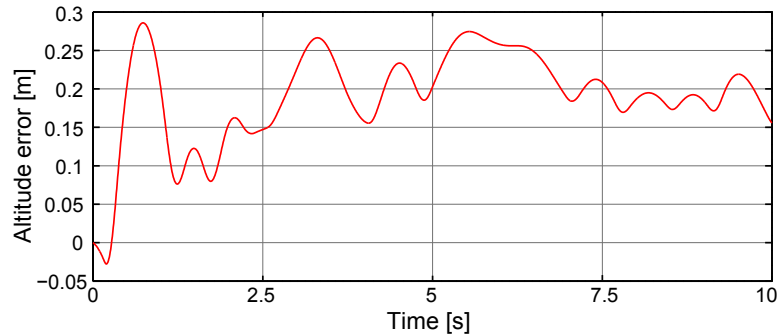


Figure 13.19: This plot shows the altitude error during of the helicopter with semi-submerged slung load towards the target position, during the 400 seconds of simulation.

Figure 13.19 depicts the altitude error during the transition phase, and it is within the limit of 0.50 meters. It is noteworthy that the altitude reference changes from 3.00 meters to 2.80 meters when going from hover to flight with the load attached, since the load is dragged behind the helicopter, and thus the helicopter needs to fly a bit lower.

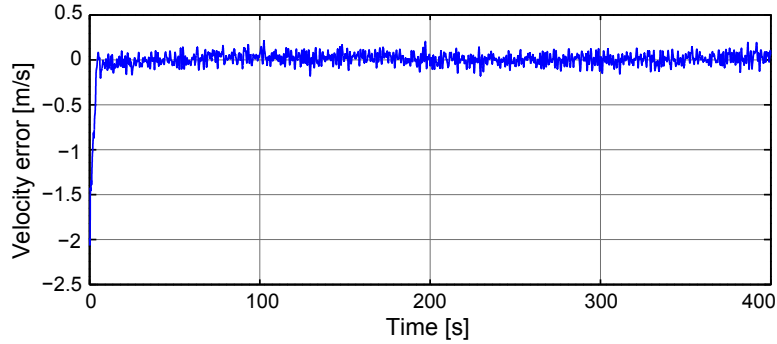


Figure 13.20: This plot shows the velocity error of the helicopter with semi-submerged slung load towards the target position, during the 400 seconds of simulation.

The largest velocity error is after the transition to flight is 0.22m/s , well within the limit of 1.00m/s . In the 10 tests of this requirement the largest velocity error is 0.30m/s and the longest transition time is 1.88 seconds, with a maximum altitude error of 0.32 meters. Table 13.9 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Velocity error	1.00m/s	0.30m/s	✓
Transition time	10.00 s	1.88 s	✓
Altitude error	1.00 m	0.32 m	✓

Table 13.9: SP2.1 results and verdict.

With MPC and LQ Control

In this test, the helicopter is set to drag the slung load to a position outside its reach within the simulation time, to observe the helicopter in flight, and the transition from hover to flight.

Figure 13.21 on the next page show the position error over the test. It is shown that the helicopter has an offset on the x-direction, which as already explained can be caused by the wind. This could probably be partly removed with larger weights on the y-velocity in the MPC, but at the cost of the startup performance.

Figure 13.22 on the facing page show the altitude error in the transition from hover to flight, where the total drop is shown to be larger than with just the LQ controller. The altitude error here is caused by the helicopter starting to drag the slung load, where the drop with LQ controller alone is caused in the moment of the controller switching.

The third requirement for the flight test is on the velocity towards the reference position. Here the results are close to those of the LQ controller, but the underlying x-velocity in the result is steadier with the MPC-LQ setup, but with larger errors in the y-velocity.

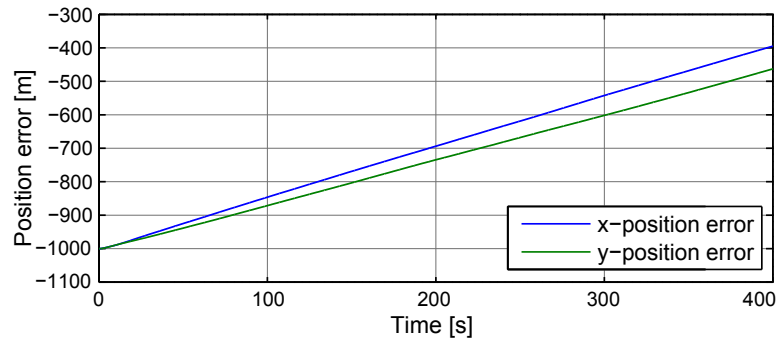


Figure 13.21: This plot shows the position error of the helicopter with load during the 400 seconds of simulation.

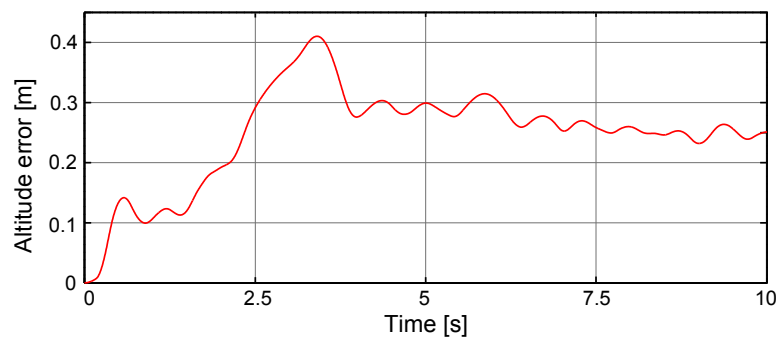


Figure 13.22: This plot shows the altitude error during of the helicopter with semi-submerged slung load towards the target position, during the 400 seconds of simulation.

The results of the test show that the MPC-LQ controller setup is able to comply with the set requirements. The worst case of the 10 simulations is a velocity error of 0.38m/s , altitude error of 0.46m with a transition time of 1.47s . The results and verdict are shown in table 13.10.

	Requirement	Result	Verdict
Velocity error	1.00m/s	0.38m/s	✓
Transition time	10.00 s	1.47 s	✓
Altitude error	1.00 m	0.46 m	✓

Table 13.10: SP2.2 results and verdict.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/SP2/...

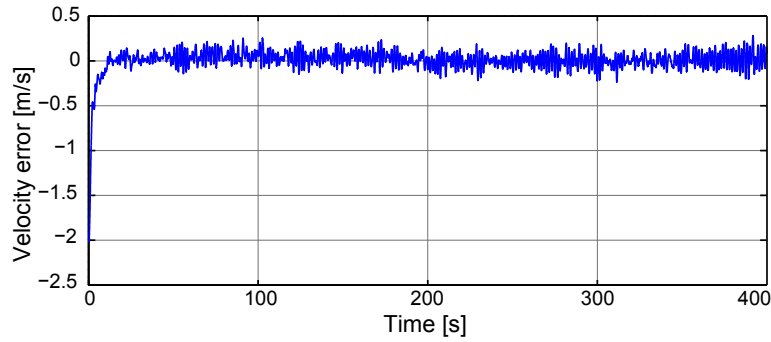


Figure 13.23: This plot shows the velocity error of the helicopter with semi-submerged slung load towards the target position, during the 400 seconds of simulation.

SS3: Preventing Fully Submerged Load, While in Flight

This requirement is all about keeping the load from being completely submerged, thus the only requirement in this acceptance test is that the load must not be completely submerged during flight. For the case of the load with the radius this means that the load must not pass $114.3 \cdot 10^{-3}$ m in altitude. The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

For this acceptance test the data from **SP2.1** is used. Figure 13.24 shows the altitude of the load during the flight, and the maximum altitude is $46.54 \cdot 10^{-3}$ m.

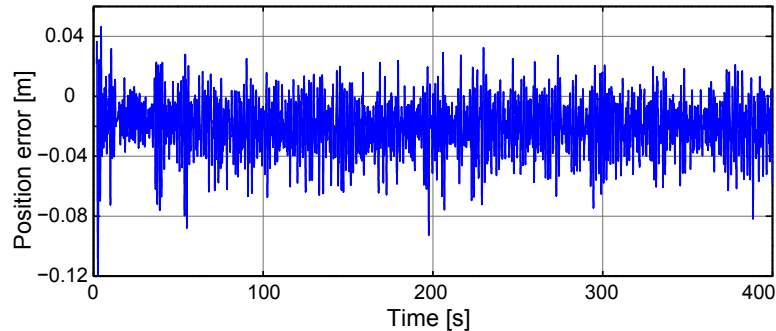


Figure 13.24: This plot shows the altitude of the load, during the flight in **SP2.1**

Of all of the 10 simulations the maximum altitude is $95.63 \cdot 10^{-3}$ m.

Table 13.11 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Altitude deviation	$114.3 \cdot 10^{-3}$ m	$95.63 \cdot 10^{-3}$ m	✓

Table 13.11: SS3.1 results and verdict.

With MPC and LQ Control

Figure 13.25 show the altitude of the slung load over the simulation of acceptance test **SP2.2**, where the slung load is being dragged through the water towards a reference position.

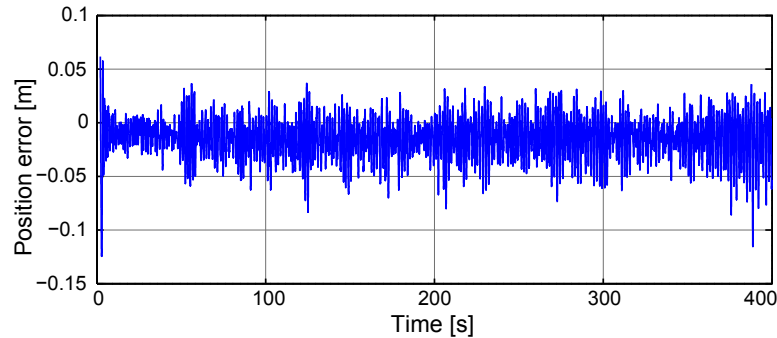


Figure 13.25: This plot shows the altitude of the load, during the flight in **SP2.2**

The results from the the 10 simulations show that the lower weightings in the previous tests have resulted in better performance in this test, with a maximum altitude deviation of the slung load of less than 0.07 m, which is well within the requirement. The figure show how the result comes from the beginning of the flight, where the helicopter start dragging the slung load. Table 13.12 show the results and verdict of the test.

	Requirement	Result	Verdict
Altitude deviation	$114.3 \cdot 10^{-3} \text{ m}$	$67.52 \cdot 10^{-3} \text{ m}$	✓

Table 13.12: SS3.2 results and verdict.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/SS3/...

SS4: Arriving at a Reference Position

The helicopter should be able fly to a specified location and switch to hover, as stated in the acceptance test specification. The transition is required to take no more than 10.00 seconds, without deviating more than 0.50 meters in altitude. Once in hover mode, it must not deviate more than 0.50 meters horizontally and vertically.

The following two sections will test if this is fulfilled with the designed LQ controller, and if it is fulfilled with MPC and LQ control.

With LQ Control

Figure 13.26 shows the horizontal position errors of the helicopter, while it flies towards the specified location, and once at the location a close-up is shown where it can be seen that it stays within the 0.50 meter boundary.

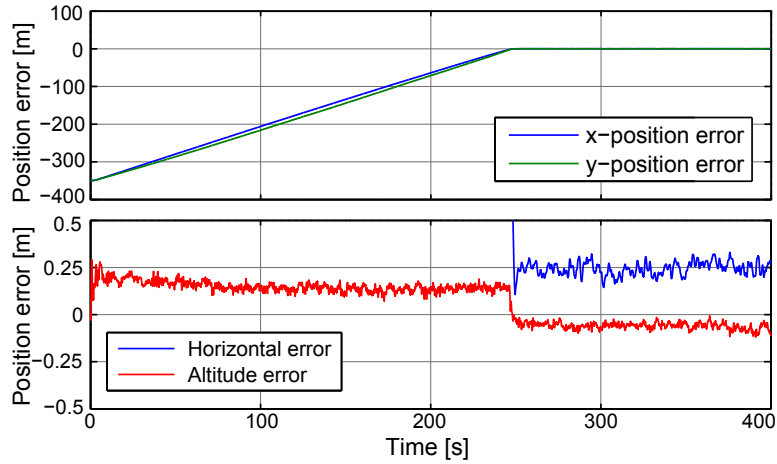


Figure 13.26: This plot shows the horizontal position, and altitude error of the helicopter during the 400 seconds of simulation. The top plot shows the horizontal positions, while the bottom plot shows a close-up on the hover region, where the altitude also is plotted.

In all of the 10 simulations the largest horizontal position error is 0.47 meters and the largest altitude error is 0.30 meters.

Figure 13.27 shows the horizontal position error and the altitude of the helicopter during the transition phase, where it can be seen that neither of them violates the 0.50 meter limit, set forth by the requirements.

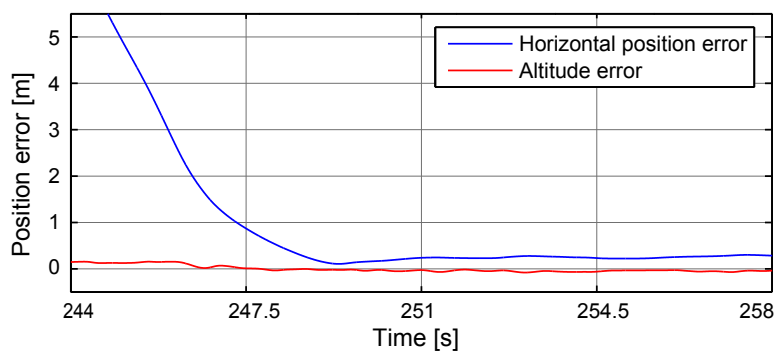


Figure 13.27: This plot shows the horizontal position and altitude error of the helicopter during the transition from flight to hover.

The transition from flight to hover, shown in figure 13.27, takes 3.21 seconds and has a maximum error in altitude of 0.16 meters. In all of the 10 simulations for this require-

ment, the slowest transition is 3.28 seconds and the largest altitude deviation is 0.23 meters.

Table 13.13 shows the verdicts on this part of the acceptance test.

	Requirement	Result	Verdict
Hover horizontal position error	0.50 m	0.47 m	✓
Hover altitude error	0.50 m	0.30 m	✓
Transition time	10.00 s	3.28 s	✓
Transition altitude error	0.50 m	0.23 m	✓

Table 13.13: SS4.1 results and verdict.

With MPC and LQ Control

In this test where the helicopter is given a reference position within the range of what can be reached during the simulation time, the transition between flight and hover is observed. Figure 13.28 show the position error over the time of the simulation, with a closeup of the position error after the reference position is reached.

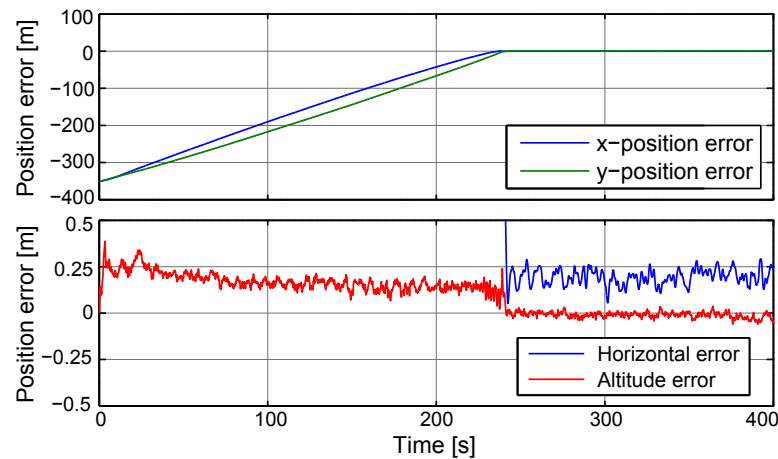


Figure 13.28: This plot shows the horizontal position, and altitude error of the helicopter during the 400 seconds of simulation. The top plot shows the horizontal positions during the entire flight, while the bottom plot shows a close-up on the hover region, where the altitude also is plotted.

Figure 13.29 on the next page show the period of the transition from flight to hover. With the helicopter pulling to the left, it comes in at a slightly larger angle than with the test with just the LQ controller, resulting in a less smooth transition.

The results from the transition test are very close to those of the test with just an LQ controller, which just as with the standalone helicopter test can be explained by the transition being handled the same way. The worst case horizontal error while in hover

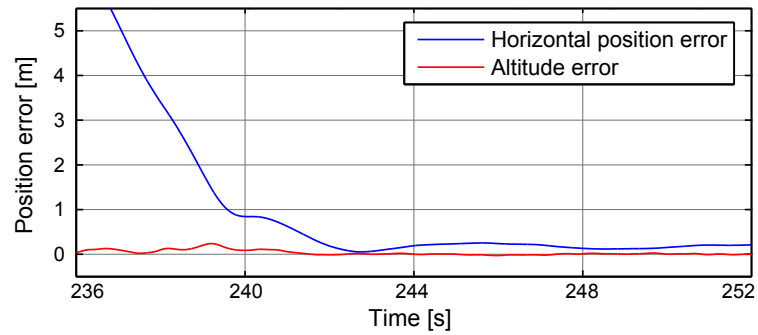


Figure 13.29: This plot shows the horizontal position and altitude error of the helicopter during the transition from flight to hover.

is 0.47 m which is very close to the requirement limit, and an altitude error of 0.21 m being well within the requirement limit.

Table 13.14 show the results and verdict of the test.

	Requirement	Result	Verdict
Hover horizontal position error	0.50 m	0.47 m	✓
Hover altitude error	0.50 m	0.21 m	✓
Transition time	10.00 s	4.45 s	✓
Transition altitude error	0.50 m	0.36 m	✓

Table 13.14: SS4.2 results and verdict.

Scripts and data from the simulations is available on the enclosed DVD:



/Acceptance Test/data/SS4/...

13.3 Acceptance test conclusion

Here some of the results from the different acceptance tests is summed up and concluded upon. This first part focus on the results for the standalone helicopter while the last part is about the helicopter with the semi-submerged slung load attached.

The results from the standalone helicopter acceptance tests are all shown in table 13.15. The hover tests revealed nearly identical results for the two control strategies. However, with the LQ being more stable, but with the MPC-LQ hovering closer to the reference position most of the time. The flight test results revealed the MPC-LQ setup to be better at controlling the velocity of the helicopter, around the set altitude. In the transition from flight to hover, both controller types had nearly identical performances, with the position errors being close to the set requirements, allowing more work on the switching between flight and hover controllers.

The MPC-LQ setup did in general have better performance than the LQ controller alone in the standalone helicopter acceptance tests, but the LQ controllers also kept the helicopter within the required limits, leaving the performance of both controller setups satisfactory.

	Requirement	LQ		MPC-LQ	
HP1 Results					
Horizontal error	0.50 m	0.37 m	✓	0.40 m	✓
Altitude error	0.50 m	0.14 m	✓	0.40 m	✓
HP2 Results					
Velocity error	1.00m/s	0.85m/s	✓	0.38m/s	✓
Transition time	10.00 s	5.78 s	✓	4.65 s	✓
Altitude error	0.50 m	0.37 m	✓	0.28 m	✓
HS3 Results					
Hover horizontal error	0.50 m	0.43 m	✓	0.47 m	✓
Hover altitude error	0.50 m	0.45 m	✓	0.41 m	✓
Transition time	10.00 s	2.97 s	✓	4.15 s	✓
Transition altitude error	0.50 m	0.46 m	✓	0.43 m	✓

Table 13.15: Result table for the standalone helicopter acceptance tests.

The second part of the acceptance tests, regarding the helicopter with a semi-submerged slung load attached, revealed little difference between the control strategies with the helicopter in hover. While in hover, both controller setups satisfied the requirements to horizontal and altitude position errors. While in flight, both controller types had similar results, but with the transition from hover to flight being more smooth with the MPC-LQ setup. The MPC-LQ setup did however also experience the biggest drop in the transition between the controllers.

With the slung load being dragged, one of the acceptance tests requirements is on the altitude of the slung load, which is not allowed to become fully submerged while in

flight. Both controller types are able to comply with the requirement, but with the MPC-LQ setup having the best results, which is a result of the altitude drop in the beginning of the flight, where without the drop, the result would have been even better.

The final acceptance test for the full system is flight to, and hover at a given reference system, where the controller transitions are observed. Here both controller types, again had nearly identical performance, however with the MPC-LQ setup having a slower transition time, which is a design choice in the MPC, where the x- and y-velocity references of the helicopter is given as an exponential function. Both controller setups are very close to the requirements, which all are complied with, some only barely.

In conclusion, both controller setups fulfilled all the requirements set to the system, and with no clear advantage of any of them, when only observing the states with requirements. Chapter 14 on the facing page compare the results of the two controller types, where the control signals, and how steady the controllers are able to hold the slung load in hover and flight, are all taken into considerations.

	Requirement	LQ		MPC-LQ	
SP1 Results					
Horizontal position error	0.50 m	0.37 m	✓	0.37 m	✓
Altitude error	0.50 m	0.29 m	✓	0.25 m	✓
SP2 Results					
Velocity error	1.00 ^m /s	0.30 ^m /s	✓	0.38 ^m /s	✓
Transition time	10.00 s	1.88 s	✓	1.47 s	✓
Altitude error	1.00 m	0.32 m	✓	0.46 m	✓
SS3 Results					
Altitude error	$114.3 \cdot 10^{-3} \text{ m}$	$95.63 \cdot 10^{-3} \text{ m}$	✓	$67.52 \cdot 10^{-3} \text{ m}$	✓
SS4 Results					
Hover horizontal error	0.50 m	0.47 m	✓	0.47 m	✓
Hover altitude error	0.50 m	0.30 m	✓	0.21 m	✓
Transition time	10.00 s	3.28 s	✓	4.45 s	✓
Transition altitude error	0.50 m	0.23 m	✓	0.36 m	✓

Table 13.16: Result table for the helicopter with slung load acceptance tests.

Comparison of the Controllers

This chapter compares the performance of the two control strategies in this project. The comparison will focus on the simulations from the acceptance test, and judge the performance in terms of both the state values, and the aggressiveness of the control signals.

14.1 Standalone Helicopter Controller Comparison

The first comparison between the controller performance is of the standalone helicopter hover test, with the results shown in figure 14.1 on the next page. Where the performance of the two controllers are shown to be nearly identical, with the difference being the references from the MPC, lowering the z-position of the helicopter. The histograms reveal how the combination of an MPC with LQ controller, make the position fluctuate more, flattening the histograms.

With the standalone helicopter in flight, figure 14.2 on the following page show the velocity components for the helicopter. While the x-velocities are close to each other, the y-velocity is shown to be steadier with the MPC-LQ setup, taking the initial curve away, where the helicopter is turning, with generated y-velocity set-points from the MPC, this may be a result of a high x-, y-velocity weighting in the MPC, allowing it to generate fast changing set-points, counteracting the wind.

Concluding on the differences between the two controllers used on the standalone helicopter, both controllers are able to perform the tasks required of them, with the LQ controller alone having the best results in hover, and the MPC-LQ controller having better performance with the helicopter in flight.

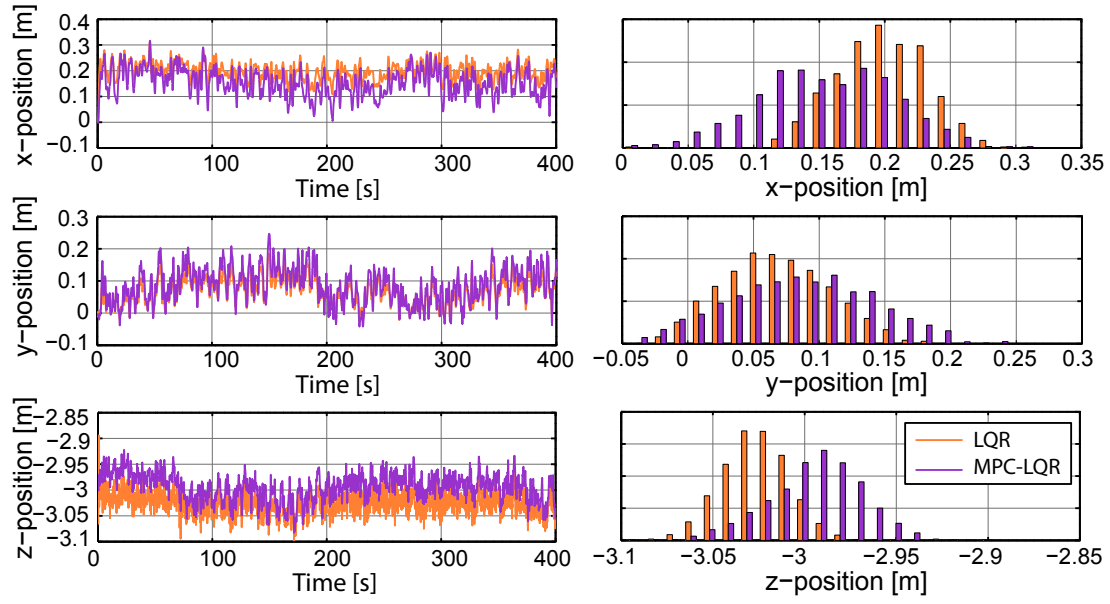


Figure 14.1: The x-, y- and z-component of the helicopter position while in hover, with histograms of all the samples.

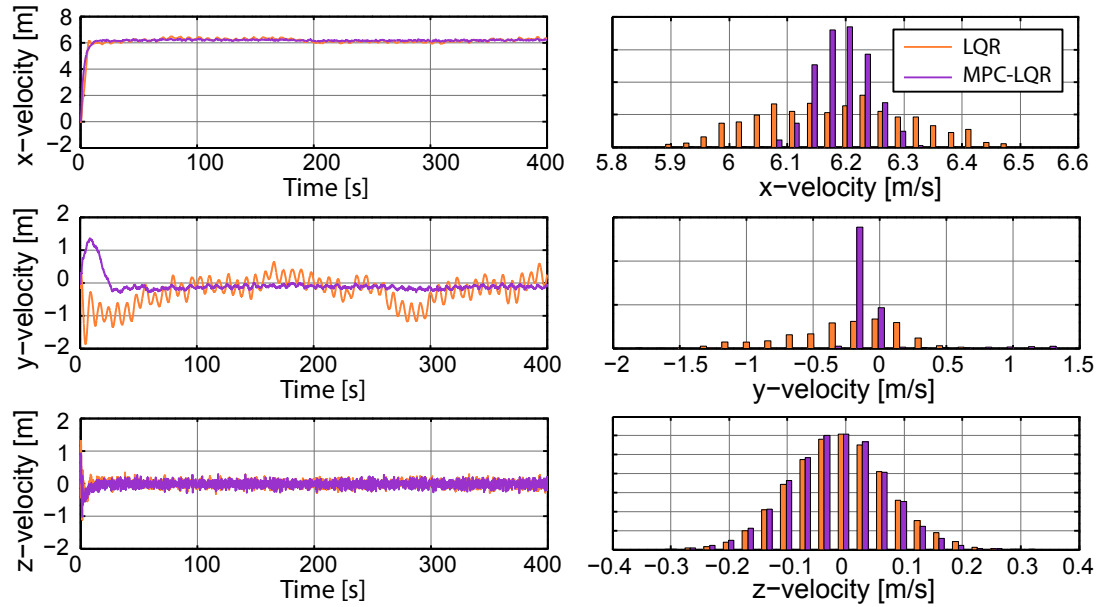


Figure 14.2: The x-, y- and z-components of the helicopter velocities while in flight, with histograms of all the samples.

14.2 Helicopter with Slung Load Controller Comparison

When comparing the controllers used on the helicopter with a semi-submerged slung load attached, the first comparison is in the hover position. Figure 14.3 show the position components with histograms. There is very little difference in the performance of the two controllers, with the only noticeable difference being the offset in the y- and z-position, which can be explained by the MPC references causing offsets for the better.

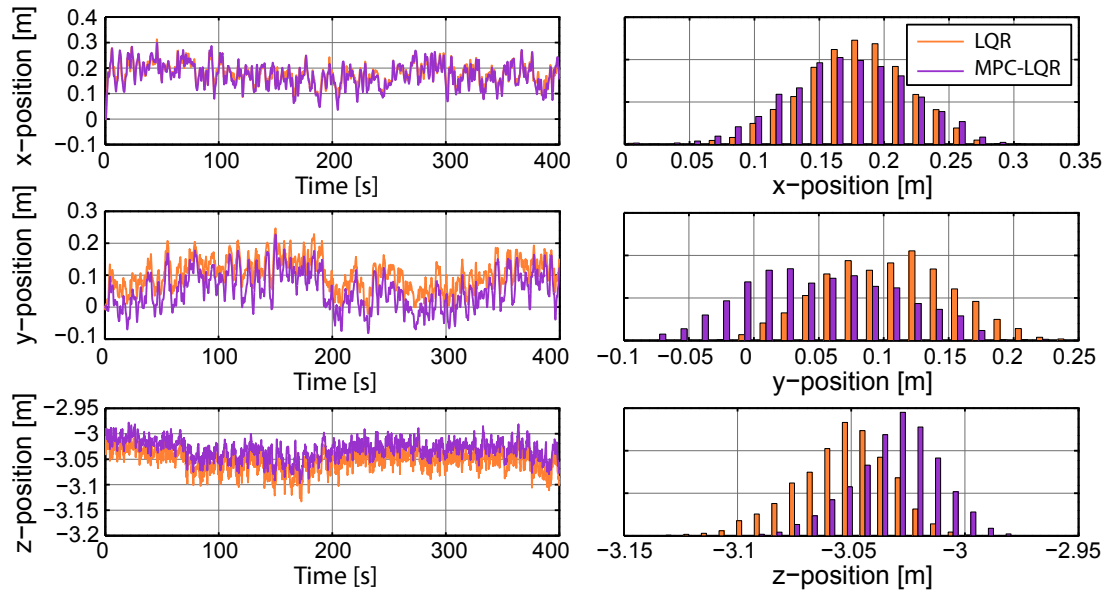


Figure 14.3: The x-, y- and z-components of the helicopter position while in hover, with histograms of all the samples.

Figure 14.4 on the following page show the control signals from the LQ controllers with the helicopter in hover, and while the differences between the controller performance in hover is small, the control signals are much more steady, this can partly be explained by the higher control signal weights used on inner loop LQ controller in the MPC-LQ control setup, and partly by the MPC generating a reference to the z-position, lowering the helicopter and load, which then require a lower collective blade pitch to maintain the altitude.

With the helicopter changed to flight, figure 14.5 on page 167 show the velocity components of the helicopter. The performances here are very close to each other after the startup, with the histograms showing nearly identical distributions. This could indicate that the MPC is not utilized to its full capabilities, and higher weightings on the velocity components in the MPC could result in better performance.

The last comparison between the two controller types take both flight, transition and hover of the helicopter with slung load into considerations. Figure 14.6 on page 167 show the Euler angles of the slung load, while the helicopter is flying to a reference position and while hovering above it. The performance of the two controllers in flight

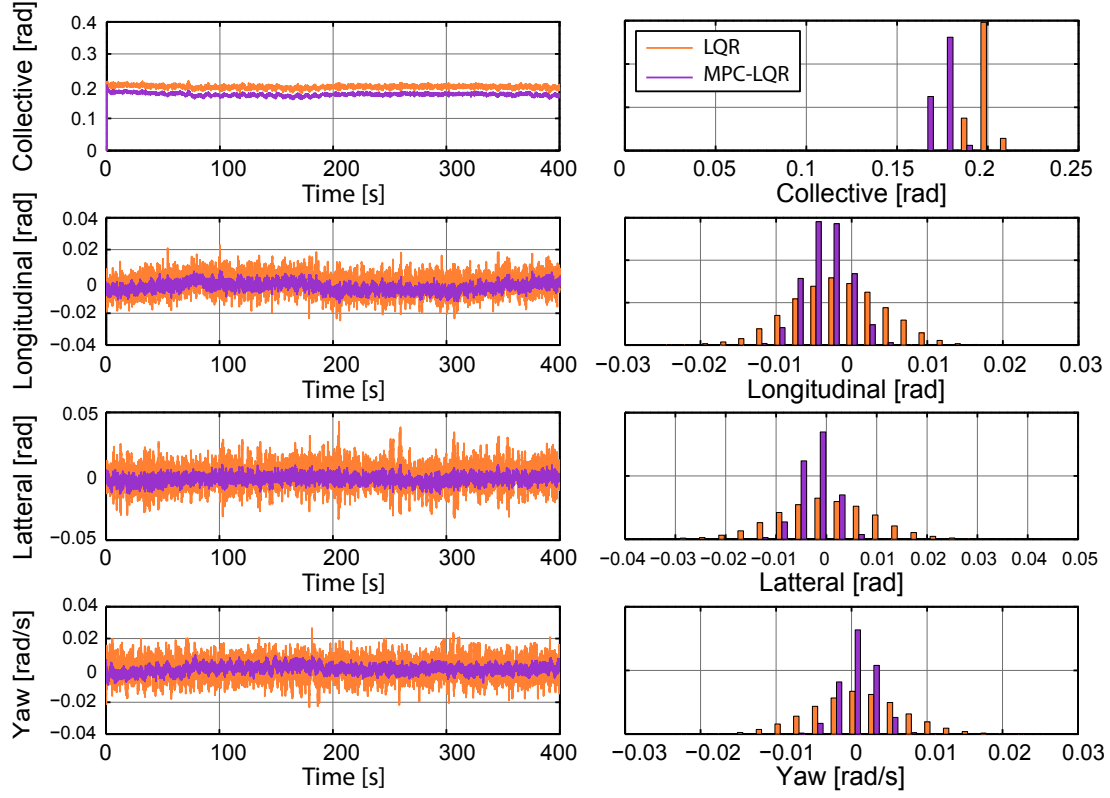


Figure 14.4: The LQ controller output signals from both controller setups, with histograms of all the samples.

are close to identical. The MPC-LQ controller is able to make a steadier transition than with just the LQ controller, but has worse performance while in hover, which is curious since only the MPC-LQ controller setup control on the slung load velocities, with the LQ controller alone depend on the slung load to be slowed down by the damping of the water.

In conclusion, both controller setups are shown to have very similar performance with the slung load attached to the helicopter. With the main difference being the MPC-LQR having steadier control signals in hover, these results could however easily be from the LQ weightings, and not necessarily because of the MPC cascade setup. The MPC-LQ setup experienced worse results for the slung load Euler angles, which could be vital when dragging an actual person through water.

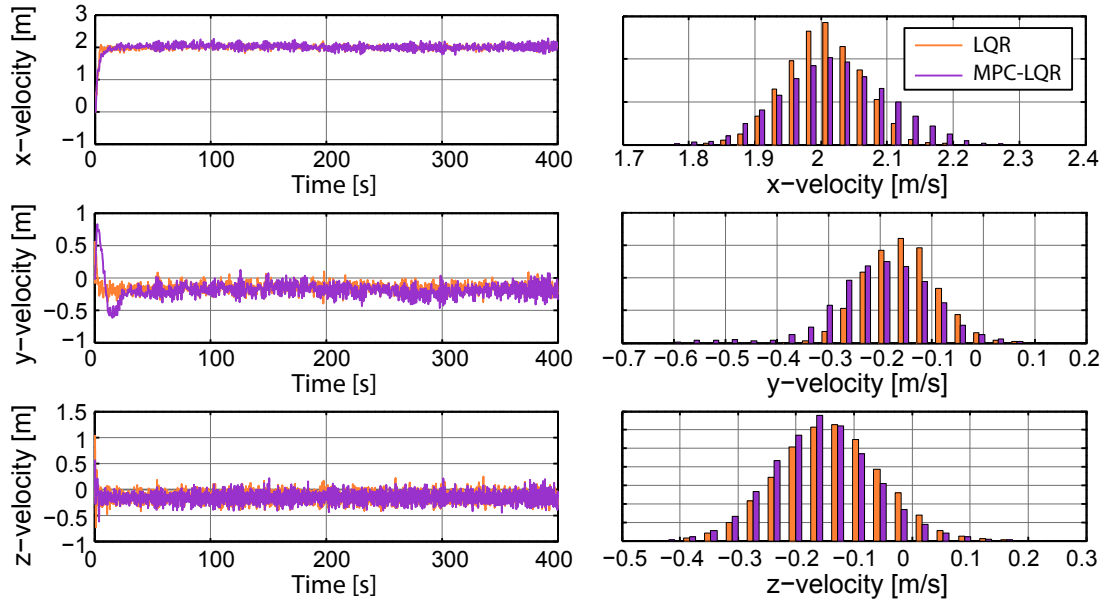


Figure 14.5: The x-, y- and z-components of the helicopter velocities while in flight, with histograms of all the samples.

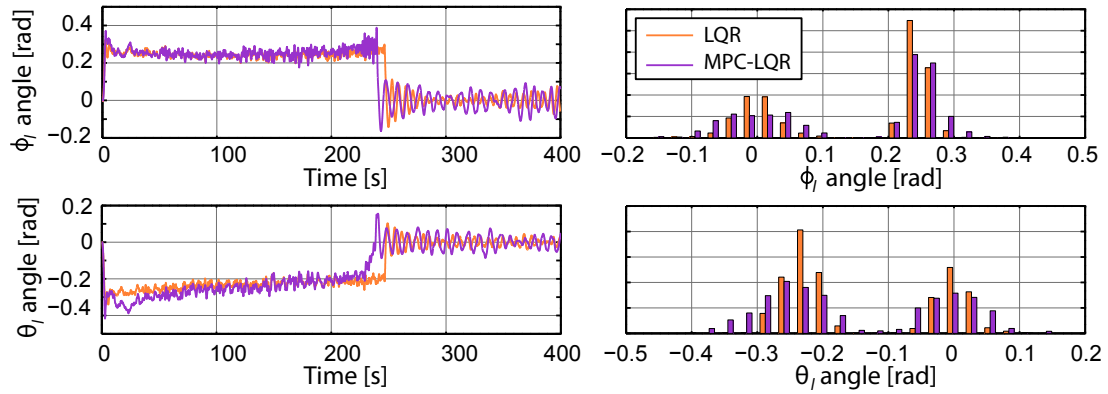


Figure 14.6: The slung load Euler angles in the SS4 acceptance test, with histograms of all the samples.

Conclusion

The first part of the report covers the introduction and motivation of the project, along with a series of answers from a SAR helicopter pilot. He specifies that the helicopters they use are very large for missions with a single distressed person, which is 95% of cases, further promoting the idea of autonomous helicopters for single person SAR missions. However, an advantage with the large helicopter is the ability to perform medical aid, after the person is taken into the helicopter, along with ability to get to shore fast after a rescue has been performed. Both being outside of the smaller autonomous helicopters reach, limiting the application options for a product like the one specified in this work, to SAR missions somewhat close to shore.

In the problem analysis the initial problem regarding an autonomous helicopter flying to, finding and dragging a person in distress, is delimited to a proof of concept system with a helicopter dragging a semi-submerged buoy through the water. Controller requirements are specified for the helicopter both with and without a slung load, to keep the: flying to, hovering at and dragging to shore, parts of the scenario.

The modelling of the semi-submerged slung load proves to be difficult, mainly due to the submerged part of the buoy not being constant, affecting the buoyancy, lift-, drag- and added mass forces. The shape of the buoy, with the submerged part seen as a spherical cap, allows for some simplifications to be made, mainly that added mass from rotations can be neglected. In the rigid body modelling chapter the modelling is even further simplified by considering the slung load as a point mass, attached to the helicopter with a nonelastic rigid wire. The rigid body model is finished with the attachment point for the slung load wire being located in the helicopters centre of mass, and thereby removing cross correlations between the helicopters rotational movement, and the movement of the slung load.

The verification and simulation part of the modelling chapter includes a simple experimental setup to check the static forces acting on the slung load, revealing a lack of sufficient damping on the system model, and a difference between the slung loads frequency while floating, and that of the slung load model. The model is concluded to be good enough to use in remaining parts of this work. The second simulation part treats the rigid body modelling, that through simulations is shown to behave as expected with the modelling simplifications that are done.

In the controller part, two control strategies are investigated. The first being LQ control for the standalone helicopter in hover and flight, and the same with the semi-submerged slung load attached. Switching is also implemented, so the helicopter would be able to switch from hover to flight and fly to a reference position, and switch back to hover. The second control strategy is model predictive control implemented in cascade as a reference controller for a inner loop LQ controller in charge of stabilising the system. Switching is also implemented for the MPC-LQ controller setup, with the chapter also including a pole analysis of the inner loop LQ controllers, where the results for the MPC-LQ setup show that the MPC has difficulties reaching large yaw angle references.

The acceptance test is split into two parts, one for the standalone helicopter and one for the helicopter with slung load. In the first part, both controller setups satisfies the requirements, where the best flight results are achieved by the MPC-LQ setup. With the slung load attached in the second part, all requirements are still complied with, but with the secondary requirements to transition between flight and hover controller being close to the maximum allowed deviations. In general both controller setups has performance close to eachother with the slung load attached. This result is also shown in the controller comparison chapter, but a difference in control strategies are shown. The LQ controller with a MPC reference controller in cascade has less fluctuations and oscillations in the control signals, than without MPC.

Further Work

The slung load modelling of this work includes the most basic of the hydrodynamic forces, with simplifications done to the added mass. Since only part of the buoy is submerged in this work, other hydrodynamic forces can be included, like current forces from waves, surface tension, and surface effect in general.

The angle dependency of the lift and drag forces are implemented by scaling between expressions found for an angle of attack at 0 rad and at $\pi/2$ rad. This method does however not reveal good results for the drag force, that with scaling is calculated too high in the range between the the expressions, compared to the reference simulations from SolidWorks. Because of this, the angle of attack dependency should be expanded to include a better fit. Since SolidWorks is used in a way that it is not intended, this leaves a possible source of error, making a verification of the simulated lift and drag forces a possible expansion option.

The rigid body modelling chapter make some modelling simplifications to delimit the work, these simplifications, being an slung load attachment point in the helicopters centre of mass, observing the slung load as a point mass with no heading, and a rigid wire connecting the slung load to the helicopter. All these modelling simplifications can be expanded to get better results from the rigid body modelling.

The control part produce a series of MPC and LQ controllers, with switching between them. The switching is however shown to only just fulfil the requirements set to it, and could thus be improved upon. The MPC type used in this work is the simplest available, with no constraints and no disturbance model, essentially leaving the MPC as an LQ controller with prediction and control horizon. Including these constraints and disturbance models in the MPC design can be the first option, if the performance of the model predictive controllers is to be improved. The original idea of using MPC, is to implement it with a flight trajectory planner, which will generate way-points, with the MPC in place to optimise the process, making a flight trajectory planner a clear expansion option of the controller setup.

The above modelling and control expansion options are all parts that can be done to improve the project, but the most evident part of further work is field testing of the helicopter with slung load attached, since the results of this work all originate from simulations, which seldom is identical to reality. This also means that the implementation of the model in the Helisim source, will have to be expanded to include the controllers as well.

Summary of part IV

The first chapter of the part included the acceptance test of this work. The results from the acceptance test included the performance of a series of MPC and LQ controllers implemented on either a standalone helicopter or a helicopter with slung load being either in hover or flight.

In the controller comparison chapter the performance of the two control strategies used in this work was compared to each other in a series of tests where both strategies were given the same task. While the LQ controllers were best at dragging the slung load smoothly, the MPC was able to hold the set-point velocities better than the LQ controller.

The conclusion went through the results and current state of this work. The motivation part included answers from a SAR pilot for background knowledge. The modelling included modelling of a semi-submerged slung load, and rigid body modelling to tie Bisgaard's helicopter model and the slung load model together. And it was concluded that the MPC and LQ controllers produced for this work worked within acceptable ranges, but with room for improvement.

In future work some of the expansion opportunities of the model, such as a better fit for the angle dependency of the lift and drag forces was suggested, since it was shown that the slung load model lacked damping, which could come from the lift and drag forces. The rigid body modelling could also be expanded, to include an attachment point away from the helicopter's centre of mass.

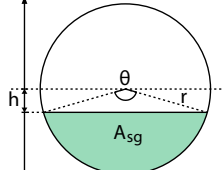
Part V

Appendices

Projected reference area

The purpose of this appendix is to obtain an expression of the reference area, which is used in the lift and drag expressions. This is a task that requires some consideration since the submerged object changes apparent shape, with the submersion depth of the buoy. For an angle of attack parallel to the water surface, the reference area can be described by the formula of a circular segment as expressed in (A.1).

$$A_{sg} = \frac{r^2 (\theta - \sin(\theta))}{2} \quad (\text{A.1})$$

A_{sg}	Area of a circular segment	m^2	
h	Distance from still water level to the centre of the sphere	m	
r	Radius of the circle (in this case the sphere)	m	
θ	Angle of the segment	rad	

Since θ can be described as: $\theta = 2 \cos^{-1}(h/r)$, it can be substituted into (A.1) to obtain a simple equation for the reference area, which is only dependent on the submersion depth.

In still water, the direction of the velocity of the buoy becomes the angle of attack as used in the Lift and drag force simulations section 6.4 on page 34. Finding an equation for the reference area becomes more complicated when the angle of attack is different than 0° , which his happens when the buoy is moving up and down in the plane of the water surface, because of the relative velocity of the incident water. An illustration of the reference area of the buoy is shown in figure A.1 on the following page, where the angle of attack is set to $\pi/4$. It is shown that the reference area is no longer a circular segment. This reference area is found by projection of the spherical segment orthogonally onto a plane with a normal vector in the direction of the relative velocity.

A MATLAB function capable of calculating this projection and plotting it is appended on the DVD:



/Simulation/Buoy/projection.m

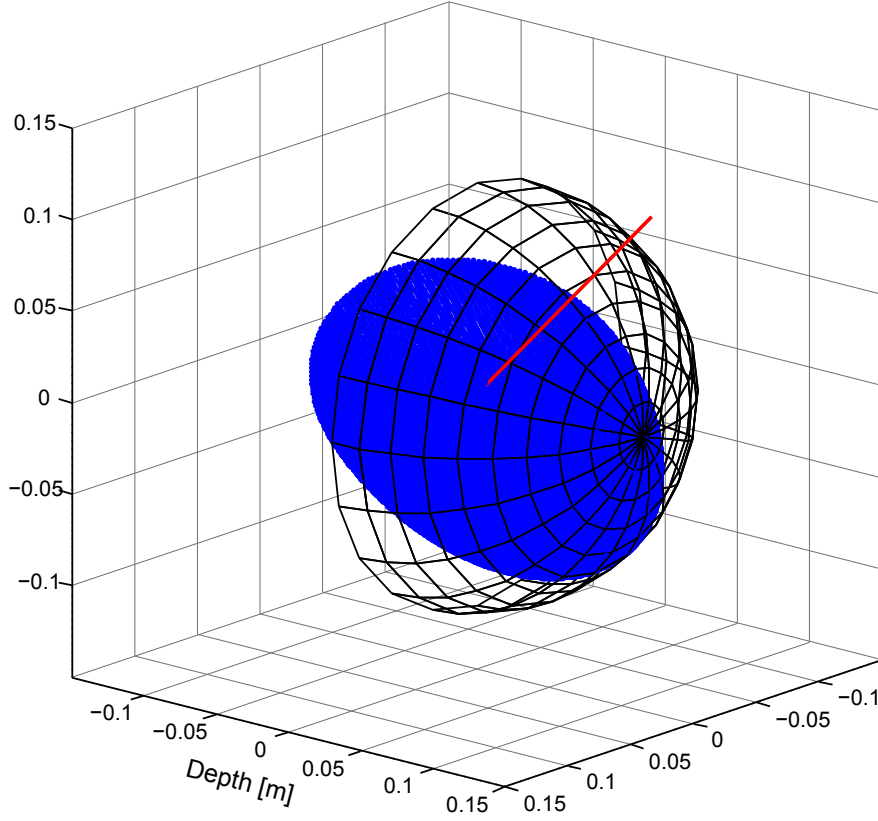


Figure A.1: A sphere segment orthogonally projected onto a plane with a normal vector in the direction of the relative velocity.

Performing this spherical segment projection for the submersion depths ranging between -0.1143 m and 0.1143 m which is the radius of the sphere, i.e. from not submerged to completely submerged, and for the angle of attack ranging between 0 and $\pi/2$, yields the plot shown in figure A.2 on the next page. The angle range between 0 and $\pi/2$ is enough because of the symmetry of the buoy.

These calculations are too heavy to run even remotely real-time, and thus an approximation is made. To gain a sufficiently close approximation to the real area, a polynomial of order 2×5 is chosen. A least squares minimisation of the polynomial provides the coefficients shown in \mathbf{P} in (A.2).

$$\mathbf{P} = \begin{bmatrix} 20.0 & 241.1 & 51.2 & -4429.0 & -321.7 & 7610.1 \\ 24.4 & -59.3 & -1889.8 & 2923.2 & 427.7 & 0.0 \\ -6.7 & -58.5 & 523.8 & 5679.7 & 0.0 & 0.0 \end{bmatrix} \cdot 10^{-3} \quad (\text{A.2})$$

\mathbf{P} Matrix containing the coefficients of the fitted polynomial

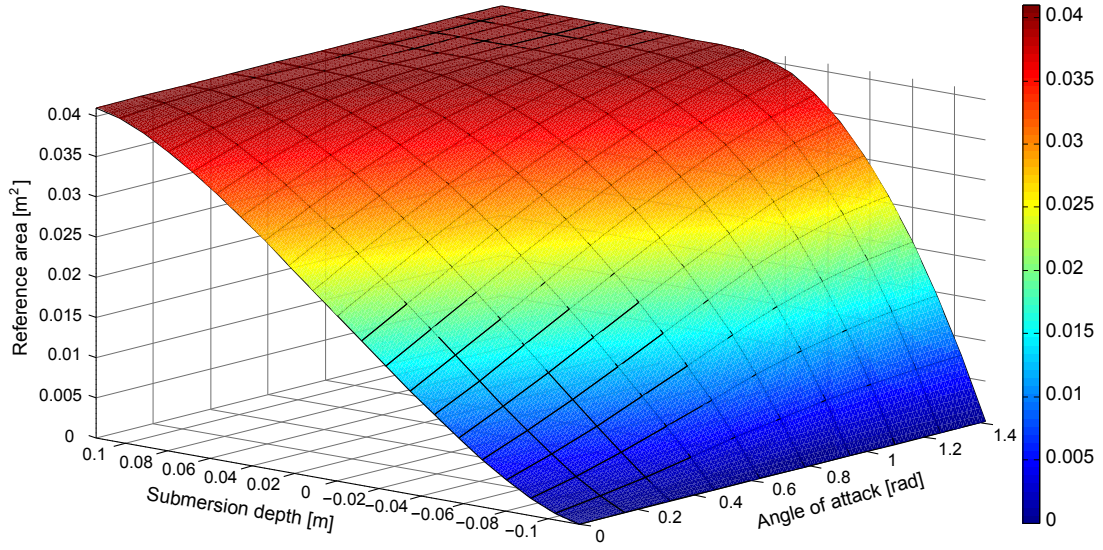


Figure A.2: Reference area as a function of angle of attack and submersion depth.

The area is the obtained by pre- and post-multiplying \mathbf{P} by the \mathbf{x} and \mathbf{y} vectors respectively as shown in (A.3).

$$A_{ref} = \mathbf{x}^T \mathbf{P} \mathbf{y} \quad (\text{A.3})$$

\mathbf{x}	Vector containing powers of the angle of attack	$[1 \ \theta \ \theta^2]^T$
\mathbf{y}	Vector containing powers of the submersion depth	$[1 \ h \ h^2 \ h^3 \ h^4 \ h^5]^T$
A_{ref}	Reference area	

Figure A.3 shows a residual plot of the fit, which reveals that the error of the fit in most of the cases is bounded below $0.5 \cdot 10^{-3} \text{ m}^2$, and in worst case bounded below $1.5 \cdot 10^{-3} \text{ m}^2$.

Considering that the largest deviations are in the most extreme angle, the fit is deemed sufficient.

A MATLAB-script capable of calculating a fit of the degrees and plotting illustrations, is appended on the DVD:



/Simulation/Buoy/projection_fit.m

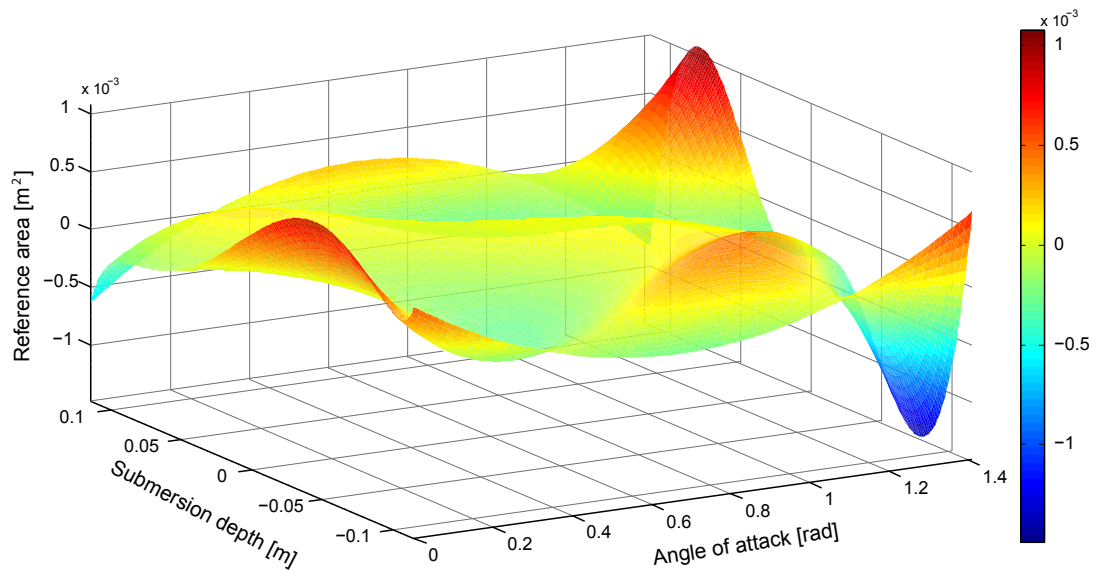


Figure A.3: Residual plot of the polynomial fit, showing maximum residuals below $1.5 \cdot 10^{-3} \text{ m}^2$.

Linearisation of Helicopter and Slung Load Model

This work uses a helicopter model developed by Morten Bisgaard in [7] at AAU. Before being able to use the model for LQ control, it needs to be linearised. Bisgaard implemented a linearisation option for the helicopter model into the source code of the project, that only require the operating point for the model to be linearised at.

B.1 Standalone Helicopter Model Linearisation

This first section only cover the linearisation of the Bisgaard helicopter model, using the linearisation option implemented in the source code of the Helisim project. However, for the `LinearizeHelicopter`-function to work, it needs a series of trimmed inputs to compute the **A** and **B** matrices at steady state. Trimming in this case mean that the control signals to the actuators are given an offset to get the system to steady state at the desired values. The helicopter trim is done using the `TrimHelicopter`-function as presented in (B.1).

$$\text{siminit} = \text{TrimHelicopter}(\text{FlightCondition}, \text{altitude}, \text{mute}, \text{save}) \quad (\text{B.1})$$

FlightCondition	For hover	[0 0 0 0]
Altitude		6.0 m
mute	suppress information	'false'
save	save trimmed values	1

$$[\mathbf{A}, \mathbf{B}] = \text{LinearizeHelicopter}(\text{siminit}, \text{steadystate}) \quad (\text{B.2})$$

siminit	Trimmed linearisation values.	
steadystate		[1 1 1 1]

The `TrimHelicopter` and `LinearizeHelicopter` functions are not only used for hover but also for forward flight where the flight conditions are changed to $[v \ 0 \ 0 \ 0]$, with v being the wanted forward flight velocity in m/s. For the flight controller in the LQ of a standalone helicopter section of the LQ chapter, a forward velocity of 6m/s is chosen. The results from the `TrimHelicopter`- and `LinearizeHelicopter`-functions for both hover and flight can be found on the DVD:



```
/Controller/lin_model_bergen_hover.mat
/Controller/Helicopter/lin_model_bergen_flight.mat
```

B.2 Helicopter with Semi-submerged Slung Load Model Linearisation

Linearised models for the complete system with a slung load attached is also needed for the controller part of this work, both for the helicopter in hover with the slung load floating, and for forward flight, with the helicopter dragging the slung load in the water surface.

Bisgaard implemented functions for trim and linearisation with a slung load attached in `TrimHelicopterAndLoad` and `LinearizeHelicopterAndLoad`. These functions did however have to be modified to use the slung load model from this work. The functions are used as shown in (B.3) and (B.4). Note that the altitude of the helicopter is not specified with this function, since it is expected to be equal to the wire length of the attached slung load, which in this work is set to 3 m.

$$\text{siminit} = \text{TrimHelicopterAndLoad}(\text{FlightCondition}, \text{mute}, \text{save}) \quad (\text{B.3})$$

$$[A, B] = \text{LinearizeHelicopterAndLoad}(\text{siminit}, \text{steadystate}) \quad (\text{B.4})$$

When using the `TrimHelicopterAndLoad` with the slung load model from this work, a problem occurred where a trim solution could not be found. This could partly be explained by the lack of proper damping on the slung load, meaning the angles of the slung load never reached a steady state, meaning that the slung load had to be set to a constant position to finish the trim process.

Two different linearised model for the helicopter with slung load system is needed, just like with the standalone case, one model for the helicopter in flight, and one for hover control. The controllers can be found on the DVD together with the trimmed values of the system.



```
/Controller/lin_model_bergen_load_hover.mat
/Controller/lin_model_bergen_load_flight.mat
```


Moving and Rotating Noninertial Frames

To get a better understanding of the math behind noninertial rotating and moving frames, as well as to be used as reference for some of the equations in the Rigid Body Model chapter 7 on page 45, an appendix dealing with the formulas of rotations and their derivatives is needed.

C.1 Moving frames

The most basic formulas are used to find the position vector for a point which position is known in another coordinate system facing the same way, while also knowing the position difference between the initial coordinate system \mathbb{E} and the moving coordinate system \mathbb{H} . The velocity and acceleration of the point can be found in the same way as the position as shown in (C.1) through (C.3).

$$\mathbb{E}\mathbf{p} = \mathbb{H}\mathbf{p} + \mathbb{E}\mathbf{p}_{EH} \quad (\text{C.1})$$

$$\mathbb{E}\mathbf{v} = \mathbb{H}\mathbf{v} + \mathbb{E}\mathbf{v}_{EH} \quad (\text{C.2})$$

$$\mathbb{E}\mathbf{a} = \mathbb{E}\mathbf{a} + \mathbb{E}\mathbf{a}_{EH} \quad (\text{C.3})$$

$\mathbb{E}\mathbf{p}, \mathbb{H}\mathbf{p}$	The position of a point given in the respective frame.	m
$\mathbb{E}\mathbf{p}_{EH}$	A vector describing the difference in position between the frames.	m
$\mathbb{E}\mathbf{v}, \mathbb{H}\mathbf{v}$	The velocity of a point described in the respective frames.	m/s
$\mathbb{E}\mathbf{v}_{EH}$	A vector describing the difference in velocity between the frames.	m/s
$\mathbb{E}\mathbf{a}, \mathbb{H}\mathbf{a}$	The acceleration of a point described in the respective frames.	m/s ²
$\mathbb{E}\mathbf{a}_{EH}$	A vector describing the difference in acceleration between the frames.	m/s ²

The frames and vectors have been named after the earth fixed frame \mathbb{E} and the helicopter fixed frame \mathbb{H} used in the rigid body modelling chapter, this is done to stay in the helicopter terminology throughout this appendix. The position vector $\mathbb{E}\mathbf{p}$ is defined from the axis unit vectors \mathbf{I}, \mathbf{J} and \mathbf{K} in the initial frame and the vector $\mathbb{H}\mathbf{p}$ defined

from the axis unit vectors \mathbf{i} , \mathbf{j} and \mathbf{k} in the moving frame as shown in figure C.1, however in this initial case with both frames having the same heading, mean that the unit vectors will be equal.

The formulas presented in (C.1) through (C.3) is only true if the moving frame have the same orientation as the inertial frame, which is seldom the case with the helicopter frame. Due to this complication, rotation of the moving frame have to be considered as well.

C.2 Moving and rotating frames

To get a better understanding of what it means that the noninertial frame \mathbb{H} is moving and rotating while seen from the inertial frame \mathbb{E} , an illustration showing some of the terms that will be used in the remaining part of this appendix is shown in figure C.1.

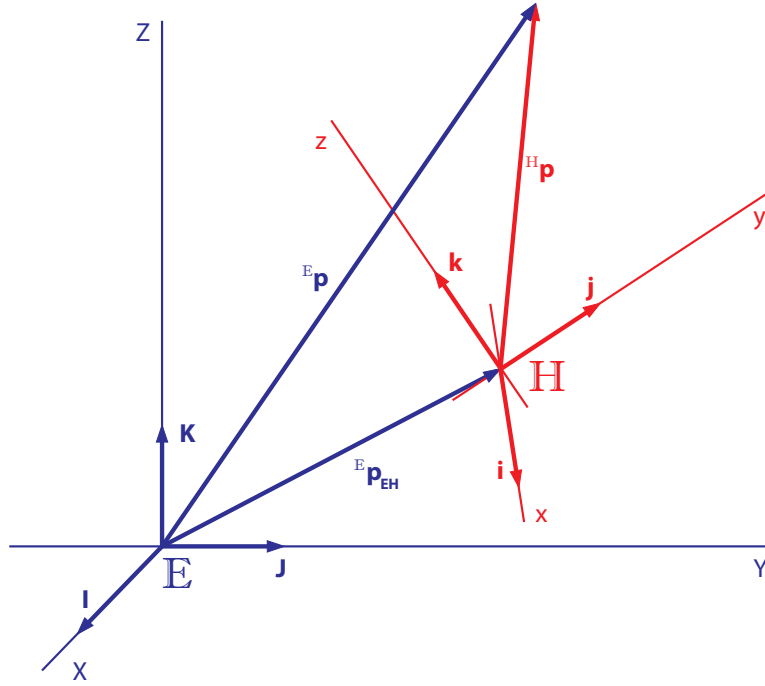


Figure C.1: An illustration showing the initial earth fixed coordinate systems noted \mathbb{E} in blue, and the moving and rotating helicopter COM fixed frame noted \mathbb{H} in red.

With the noninertial frame rotated, it is still possible to find the position of a point in the inertial frame as shown earlier in (C.1) if the position vector ${}^{\mathbb{H}}\mathbf{p}$ is rotated to fit the orientation of the \mathbb{E} frame using a rotation matrix ${}^{\mathbb{E}}\mathbf{R}_{\mathbb{H}}$ as shown in (C.4). The formulas for velocity and acceleration is however no longer true, due to the rotation illustrated in figure C.2 on the next page.

$$\mathbb{E}\mathbf{p} = \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p} + \mathbb{E}\mathbf{p}_{EH} \quad (\text{C.4})$$

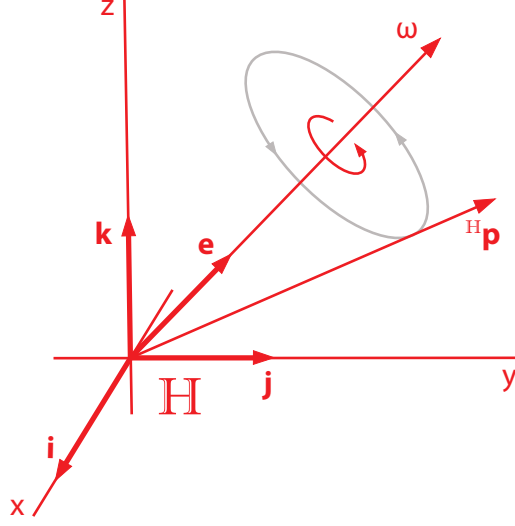


Figure C.2: An illustration rotating frame \mathbb{H} with the axis of rotation $\boldsymbol{\omega}$ included.

The newly introduced rotation have to be included in the new formula for velocity, this can be done by including a term for rotational velocity $\mathbb{E}\boldsymbol{\omega} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p}$ into the already existing formula for the velocity as done in (C.5). The rotation matrix $\mathbb{E}\mathbf{R}$ is used to transform the vectors defined in the \mathbb{H} frame into the \mathbb{E} frame.

$$\mathbb{E}\mathbf{v} = \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{v} + \mathbb{E}\boldsymbol{\omega} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p} + \mathbb{E}\mathbf{v}_{EH} \quad (\text{C.5})$$

$$\mathbb{E}\boldsymbol{\omega} = \omega \mathbb{E}\mathbf{e} \quad (\text{C.6})$$

$\mathbb{E}\boldsymbol{\omega}$	The angular velocity of the moving system.	rad/s
$\mathbb{E}\boldsymbol{\omega} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p}$	Term known as the <i>Rotational velocity</i>	
$\mathbb{H}\mathbf{e}$	A unit vector specifying the axis of rotation.	
ω	Angular speed about the axis specified by $\mathbb{H}\mathbf{e}$	

While knowing how to include the rotation in the time derivative of the position to find the velocity, this can be applied again to find the acceleration, yielding the result shown in (C.7). Also here is the rotation matrix $\mathbb{E}\mathbf{R}$ needed to transform all terms into the \mathbb{E} frame.

$$\mathbb{E}\mathbf{a} = \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{a} + 2\mathbb{E}\boldsymbol{\omega} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{v} + \mathbb{E}\boldsymbol{\alpha} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p} + \mathbb{E}\boldsymbol{\omega} \times (\mathbb{E}\boldsymbol{\omega} \times \mathbb{E}\mathbf{R}^{\mathbb{H}}\mathbf{p}) + \mathbb{E}\mathbf{a}_{EH} \quad (\text{C.7})$$

${}^{\mathbb{E}}\boldsymbol{\alpha}$	The angular acceleration of a moving system.	rad/s^2
$2{}^{\mathbb{E}}\boldsymbol{\omega} \times {}^{\mathbb{E}}\mathbf{R}^{\mathbb{H}}\mathbf{v}$	Term known as the <i>Coriolis acceleration</i> .	
${}^{\mathbb{H}}\boldsymbol{\alpha} \times {}^{\mathbb{H}}\mathbf{R}^{\mathbb{H}}\mathbf{p}$	Term known as the <i>Transverse acceleration</i> .	
${}^{\mathbb{E}}\boldsymbol{\omega} \times ({}^{\mathbb{E}}\boldsymbol{\omega} \times {}^{\mathbb{E}}\mathbf{R}^{\mathbb{H}}\mathbf{p})$	Term known as the <i>Centripetal acceleration</i> .	

C.3 Example: Rotation and velocity difference

A basic example showing some of the principles from this appendix, is a point located in a frame rotating with a constant velocity, and then observing that point in both the rotating and the initial coordinate system, corresponding to the illustrations in figure C.1 on page 182 and C.2 on the previous page with a constant ω .

The rotation of the \mathbb{H} frame is around the x-axis in this example, giving a rotation matrix ${}^{\mathbb{E}}\mathbf{R}$ as shown in (C.8), and a velocity difference between the two frames have been set in the x-axis for illustrative purposes. The position of a point in the rotating frame displayed by a vector ${}^{\mathbb{H}}\mathbf{p} = [0\ 1\ 0]^T$ m, with the rotating frame having a constant velocity of $\omega = [0.5\ 0\ 0]^T$ rad/s and a velocity difference between the two frames ${}^{\mathbb{E}}\mathbf{p}_{EH} = [0.1\ 0\ 0]^T$ m/s. The results are the position of the vector seen from the initial frame \mathbb{E} , illustrated in figure C.3.

$${}^{\mathbb{E}}\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (\text{C.8})$$

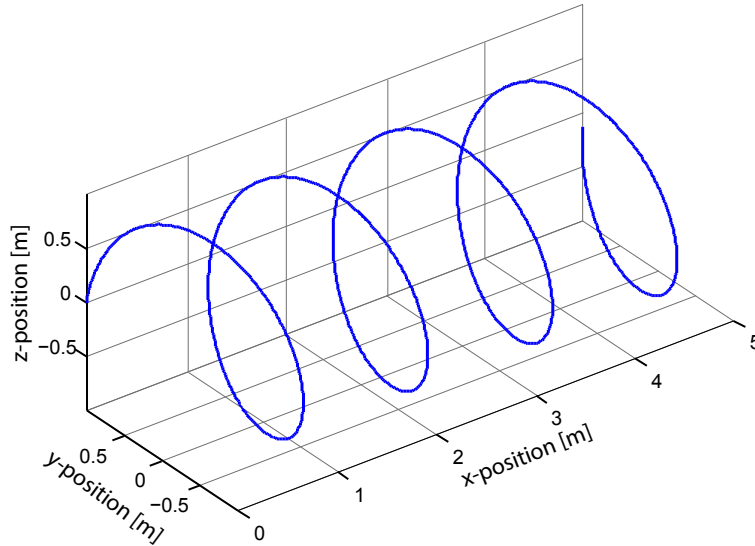


Figure C.3: A 3D plot of the vectors position seen from the initial frame \mathbb{E} .

So when observing the vector in the rotating frame it is constant, but when observing it from the initial frame it is rotating in the yz-plane, and moving slowly away due to

the difference in velocity of the two frames. Figure C.4 show the y-composant of the ${}^{\mathbb{H}}\mathbf{p}$ vector, and the resulting y- and z-composants in the ${}^{\mathbb{E}}\mathbf{p}$ vector.

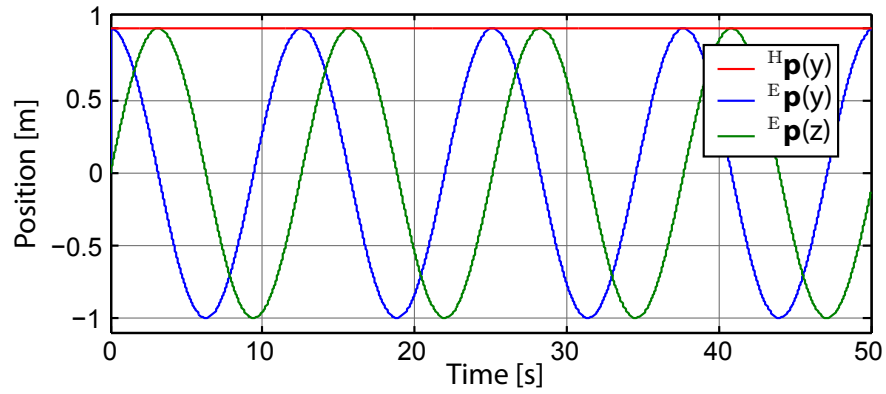


Figure C.4: A plot of the vectors y-composant in the rotating frame \mathbb{H} and the resulting y- and z-composant observed in the \mathbb{E} frame.

The MATLAB-script have been included on the DVD:



/Model/Rigid body model/rotating_example_1.m

Bibliography

- [1] P. Andersen. Optimal control, February 2010.
- [2] D. G. Antonelli. *Underwater Robots - Motion and Force Control of Vehicle-Manipulator Systems*, volume Second. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, June 2006.
- [3] aqua calc. Human body.
- [4] S. Archives. First helicopter naval rescue.
- [5] M. Bernard and K. Kondak. Generic slung load transportation system using small size helicopters. *IEEE*, 2009.
- [6] M. Bisgaard. Modeling, estimation, and control of helicopter slung load system. *IEEE*, 2007.
- [7] M. Bisgaard. *Modeling, Estimation, and Control of Helicopter Slung Load System*. PhD thesis, Aalborg University, 2007.
- [8] E. F. C. . C. Bordons. *Model Predictive Control*. Springer, second edition, 2003.
- [9] G. S. Bla Liptk, Greg McMillan and H. Wade. Control, a gathering of eagles: Ask the experts special edition. *Control Magazine*, January 2010.
- [10] G. G. D. Fusato. Flight dynamics of an articulated rotor helicopter with an external slung load. *Journal of the American Helicopter Society*, 46:3–13, 2001.
- [11] F. El-Hawary, editor. *The Ocean Engineering Handbook*. CRC Press LLC, 2001.
- [12] G. L. C. Grant R. Fowles. *Analytical Mechanics*. Brooks/Cole, Cengage Learning, seventh edition, 2005.
- [13] A. U. U. group. The bergen helicopter, September 2011.
- [14] U. S. C. Guard. Summary statistics 1964 thru 2006.
- [15] T. L. Heath. The works of arhimedes. *Cambridge University Press Warehouse*, page 257, March 1897. Proposition 5.
- [16] J. M. Maciejowski. *Predictive Control with Constraints*. Pearson, 2002.
- [17] W. A. museum. Vergulde draak (draeck) (gilt dragon) wreck.
- [18] K. T. Palle Andersen. Model predictive control lecture notes.

- [19] R. Stuckey. Mathematical modelling of helicopter slung-load systems. *IEEE*, 2001.
- [20] A. J. Srensen. Marine control systems - propulsion and motion control of ships and ocean structures.
- [21] P. A. Techet. 2.016 hydrodynamics.