

FMH606 Master's Thesis 2017

Master of Science, Industrial IT and Automation

Modelling and Control of Two Degrees of Freedom Helicopter Model

Sondre Nilsen

The University College of Southeast Norway takes no responsibility for the results and conclusions in this student report.

Course: FMH606 Master's Thesis, 2017

Title: Modelling and control of two degrees of freedom helicopter model

Number of pages: 40 report + 2 appendices = 42

Keywords: Two degrees of freedom helicopter, MIMO System, PID.

Student: Sondre Nilsen
Supervisor: Roshan Sharma
External partner: None
Availability: Open

Approved for archiving: _____
(supervisor signature)

Summary:

USN has built multiple two degrees of freedom helicopters models that are to be used in various courses at USN to expand the knowledge of modeling and control. The helicopter model is a highly unstable system, making it a challenging control object to be used in the various courses at USN.

A mathematical model is to be developed for simulation and controller development. Parameters for the model shall be found by conduction experiments on the real system. PID and MPC controllers are to be designed based on the mathematical model.

The mathematical model developed using the parameters found by conducting experiments on the real system fits the real system well, and was successfully used to develop PID controllers for the system. The developed PID controller was able to control the real helicopter model to setpoint within reasonable time and kept the pitch angle at setpoint with intermittent movement about 3° off setpoint because of random disturbances.

The model was linearized and a Kalman filter was tuned for the MPC controller, but, due to unforeseen problems with Simulink and lack of time, no MPC controller was developed.

Preface

This report is a Master Thesis written by a student of Industrial IT and Automation at University College of South East-Norway (USN). It is written in the 4th semester and presents the work done in modeling and control of a two degrees of freedom helicopter model during the spring of 2017.

Some knowledge of modeling and control theory is necessary to fully understand the report.

I would like to thank my advisor Roshan Sharma for valuable advice and guidance throughout the project and Fredrik Hansen for help with the physical model.

The following software was utilized throughout the project:

- MATLAB R2016b
- MS. Office

Porsgrunn, 07.08.2017

Sondre Nilsen

Contents

Preface	3
Contents.....	4
1..Introduction	5
1.1 Background	5
1.2 Task Description	6
2..System Description.....	7
2.1 Physical System.....	7
2.2 Forces Acting on the System	7
2.3 System Dynamics	8
3..Previous Research on Two Degrees of Freedom Helicopter models.....	10
3.1 Euler–Lagrange Equation Model, PID and MPC Controllers	10
4..Dynamic Model of two Degrees of Freedom Helicopter Model	11
4.1 Euler-Lagrange Model Formulation	11
5..Calculating Model Parameters using the Real System	14
5.1 Calculating Mass of the Arm	14
5.1.1 <i>Mass Calculation with Center of Mass at 40 mm</i>	15
5.1.2 <i>Mass Calculation with Center of Mass at 15 mm</i>	15
5.1.3 <i>Reasons for Difference in Mass Calculation</i>	15
5.2 Data Acquisition using Simulink.....	15
5.3 Calculating Moment of Inertia	16
5.4 Calculating of Rotor Gain	19
5.5 Calculating Friction Factor $\beta\phi$	20
5.5.1 <i>Finding Friction Factor $\beta\theta$</i>	22
5.6 Building the Model in Simulink	22
5.7 Model Tuning.....	24
5.8 Complete Model with all Parameters Quantified	25
6..PID Control	26
6.1 PID Controller on Velocity Form	26
6.2 PID Parameters	27
7..MPC Control	32
7.1 Linear Model	32
7.1.1 <i>Linearization</i>	32
7.1.2 <i>Linearized Model</i>	33
7.2 Kalman Filter	34
7.3 LQ Optimal Control.....	37
8..Conclusion	38
References.....	39
Appendices.....	40

1 Introduction

This chapter presents the background for the Master's thesis.

1.1 Background

Two degrees of freedom helicopters is a popular modeling and control platform due to its highly non-linear nature and is much used in research and teaching environments. The University College of Southeast Norway(USN) have built multiple two degrees of freedom helicopter models to be used in the Master's program in Industrial IT and Automation. The goal is to use the model and the controllers in various courses the at USN.

The modeling and control tools used can be applied to many similar control areas in aerospace with position control of helicopters and airplanes to marine industries such as dynamic position control for ships.

The helicopter model built by USN is seen in Figure 1-1 and is 36 cm from the bottom to the top where the arm is connected. The arm measures 56 cm from the end of one rotor to the end of the other rotor. The box seen on the left contains the power supply, the DAQ, controller used for converting the incremental position sensor signal to a position signal and the speed controllers for the rotors.



Figure 1-1: Picture of the two degrees of freedom helicopter model built at USN.

1.2 Task Description

The report covers the development of a mathematical model based on the physical model built by USN. Experiments are to be conducted to calculate the parameters in the model such that the model can be implemented in Simulink. PID and MPC controllers shall be developed for testing on the model and the real system.

2 System Description

This chapter describes the system and the system dynamics.

2.1 Physical System

The helicopter model is a Twin Rotor Multiple-inputs Multiple-outputs System. A representation of the physical model to be used is shown in Figure 2-1. Seen in Figure 2-1 is the yaw rotor to the left, the pitch rotor to the right. The rotors are of equal size. The model has two degrees of freedom, pitch shown in green and yaw shown in red. Also shown is the black incremental position sensor for the pitch position.

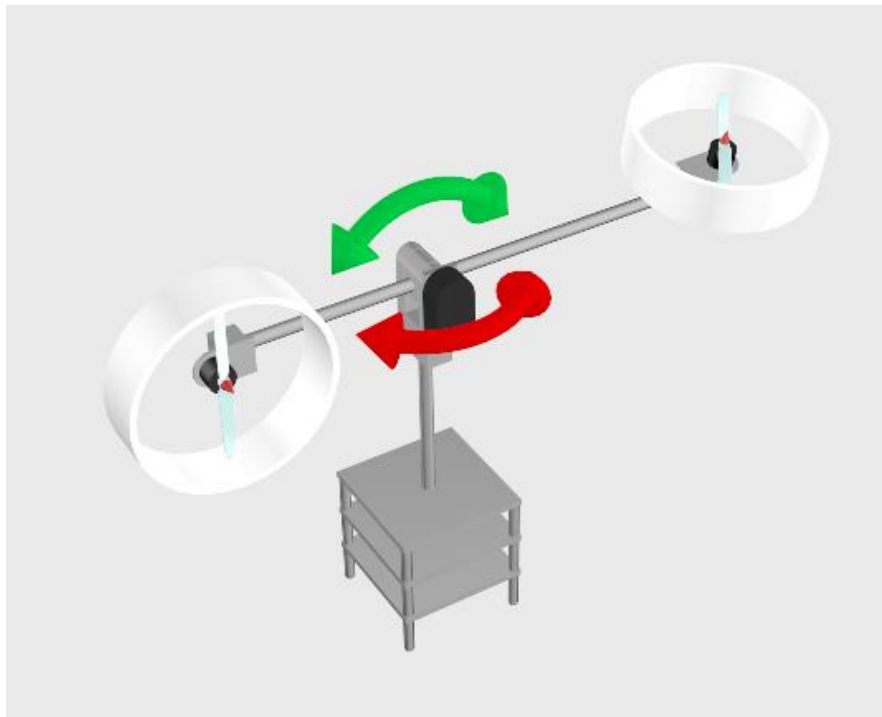


Figure 2-1: The helicopter model with two degrees of freedom. Green pitch movement and red yaw movement.

The incremental sensor for the yaw position is positioned at the bottom of the vertical pipe between the bottom two plates. The movement in yaw is restricted to the ranges -90° to 90° , giving it a span of 180° . The position sensor gives the position in volts in the range 0 to 5 V. The movement in pitch is restricted to the ranges from -48° to 40° and the position sensor gives the position in volts from 0 to 4 V.

The pitch rotor is tilted approximately 35° to achieve the yaw force found in real helicopters.

2.2 Forces Acting on the System

There are several forces acting on the system. The two propellers, gravity and friction forces.

In Figure 2-2, the red arrows point in the direction of the forces of the pitch rotor and the blue arrow in the direction of the force of the yaw rotor. The green arrow is the resulting force of

2 System Description

gravity on the arm of the helicopter model. Not shown are the friction forces that acts against the direction of the movement in the joints.

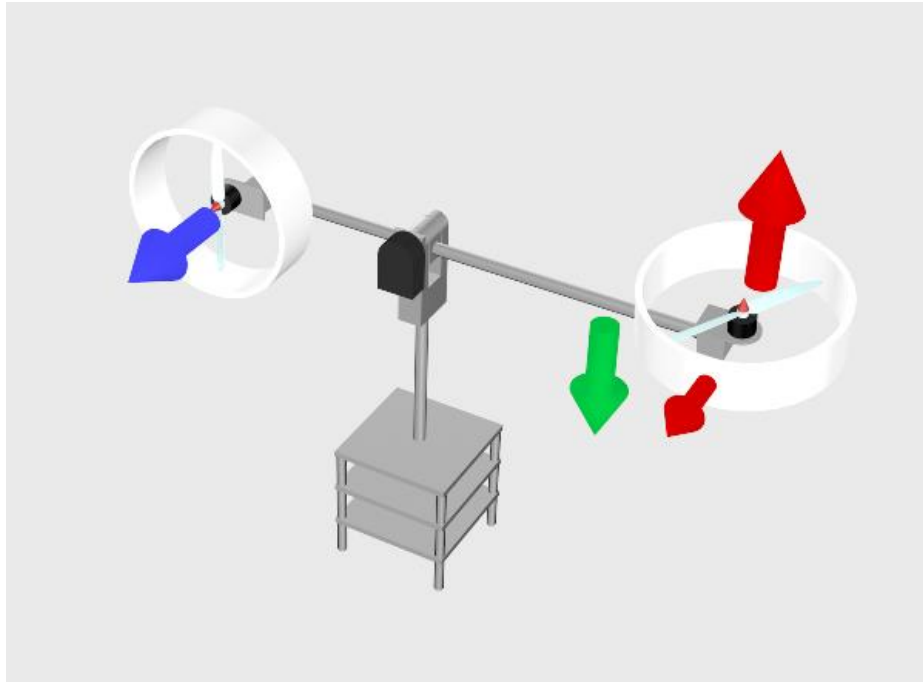


Figure 2-2: Forces acting on the system. Green illustrates the force by gravity, blue the force produced by the yaw rotor and red the force by the pitch rotor.

The rotors are controlled by a voltage signal from 0 to 5 V.

2.3 System Dynamics

The torques produced by the forces acting on the system depend on several factors. The torque produced by gravity depends on the distance between the rotational axis and the center of gravity of the arm, the pitch angle and the weight of the arm with the rotors. The torque produced by the force of the pitch and yaw rotors depends on the distances from the axes to the rotors.

The system has moments of inertia around the horizontal and vertical axes. The moments of inertia and the combined resulting torques by the forces in the system, gives the angular acceleration of the system.

The system is highly unstable in pitch angle and with significant oscillations when the control inputs kept constant, as seen in Figure 2-3.

2 System Description

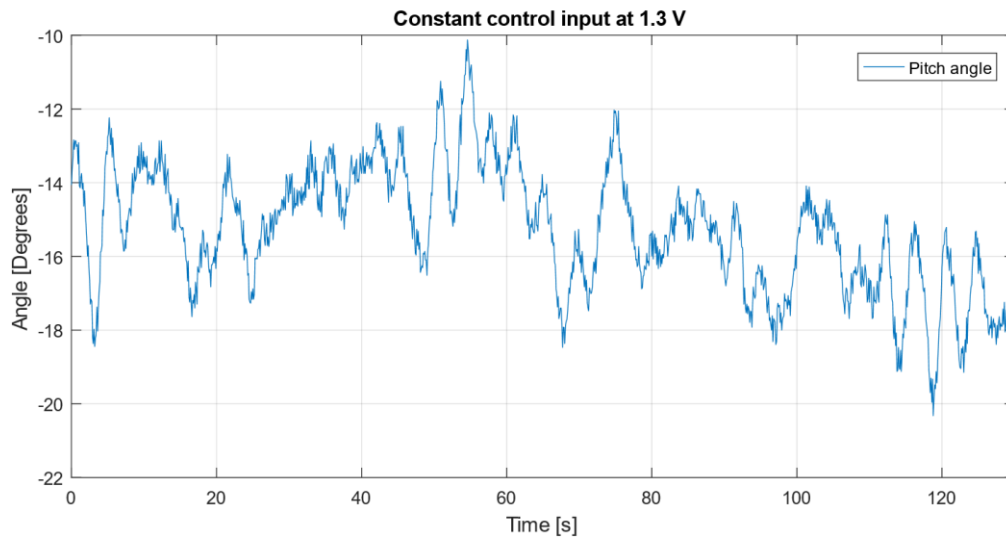


Figure 2-3: With both control inputs for the rotors kept constant, the arm of the helicopter oscillates at random with 10° from the highest to the lowest point.

3 Previous Research on Two Degrees of Freedom Helicopter models

Multiple studies have been conducted on mathematical modeling and control of two degrees of freedom helicopters. This chapter briefly presents two relevant studies used for inspiration.

3.1 Euler–Lagrange Equation Model, PID and MPC Controllers

Gutierrez, J. A, Duarte, E and Arcos, J describes in the conference paper: CONTROL SYSTEM FOR A TWO DEGREES OF FREEDOM HELICOPTER[1] how the Euler-Lagrange equation is used to develop a non-linear model of the system. The non-linear model is linearized to a state space model with 4 states to be used in a MPC controller called a servotracking controller with state observer in the paper. The article presents results comparing the servotracking controller with state observer to PID controllers and the servotracking controller performs quicker control than the PID controllers.

In the article: Model predictive control of 2DOF helicopter[2], by A.P.S. Ramalakshmi, P.S. Manoharan, K. Harshath, and M. Varatharajan a model developed on the Euler-Lagrange equation is used to describe the system. The model is implemented in a MATLAB simulation model and the model controlled by a controller designed using the MATLAB MPC toolbox. The MPC controller is compared to a PID controller used and the conclusion is that the MPC controller performs better control than the PID controller.

4 Dynamic Model of two Degrees of Freedom Helicopter Model

4.1 Euler-Lagrange Model Formulation

A dynamic model of the system is developed using the Euler-Lagrange equation, where the Lagrangian L is defined as the Kinetic energy T minus the Potential energy V of the system as show in equation (4.1).

$$L = T - V \quad (4.1)$$

The Lagrangian equation defined in

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j \quad (4.2)$$

Where:

- q_j = Generalized coordinates
- Q_j = Generalized forces

The model has two degrees of freedom and has therefore two generalized coordinates. The coordinates are the pitch θ and yaw ϕ angles in radians.

Inserting equation (4.1) into (4.2) expands the equation to equation (4.3).

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_j} \right) - \frac{d}{dt} \left(\frac{\partial V}{\partial \dot{q}_j} \right) - \frac{\partial T}{\partial q_j} + \frac{\partial V}{\partial q_j} = Q_j \quad (4.3)$$

Defining the potential energy of the system V in equation (4.4).

$$V = mgh = mgd_{cm}(1 + \sin(\theta)) \quad (4.4)$$

Where:

- d_{cm} = Distance from rotational axis pitch to center of mass of the arm
- g = Gravity constant
- m = Mass of arm
- θ = Pitch position

Defining the kinetic energy of the system T in equation (4.5).

$$T_j = \frac{1}{2} I_i \omega_j^2 = \frac{1}{2} I_\theta \dot{\theta}^2 + \frac{1}{2} I_\phi \dot{\phi}^2 \quad (4.5)$$

Where:

4 Dynamic Model of two Degrees of Freedom Helicopter Model

- I_i = Moment of inertia around rotational axis i , simplified as θ and ϕ , the generalized coordinates
- ω = Angular velocity
- ϕ = Yaw position

The moment of inertia I_ϕ is a function of θ . The distance to the various parts of the arm from the rotational axis varies with the angle θ . However, the model for I_ϕ is simplified and is assumed to be constant, and the same as I_θ as seen in equation (4.6).

$$I_\phi(\theta) \approx I_\phi \approx I_\theta \quad (4.6)$$

Expanding the different terms in equation (4.3) with respect to equation (4.4), (4.5) and (4.6).

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\theta}} \right) = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{\theta}} \left(\frac{1}{2} I_\theta \dot{\theta}^2 + \frac{1}{2} I_\phi \dot{\phi}^2 \right) \right) = \frac{d}{dt} I_\theta \dot{\theta} = I_\theta \ddot{\theta} \quad (4.7)$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) = \frac{d}{dt} \left(\frac{\partial}{\partial \dot{\phi}} \left(\frac{1}{2} I_\theta \dot{\theta}^2 + \frac{1}{2} I_\phi \dot{\phi}^2 \right) \right) = \frac{d}{dt} I_\phi \dot{\phi} = I_\phi \ddot{\phi} \quad (4.8)$$

$$\frac{d}{dt} \left(\frac{\partial V}{\partial \dot{q}_j} \right) = 0 \quad (4.9)$$

$$\frac{\partial T}{\partial q_j} = 0 \quad (4.10)$$

$$\frac{\partial V}{\partial \theta} = \frac{\partial}{\partial \theta} mgd_{cm}(1 + \sin(\theta)) = mgd_{cm} \cos(\theta) \quad (4.11)$$

$$\frac{\partial V}{\partial \phi} = 0 \quad (4.12)$$

The right side of the Lagrangian equation (4.2) is the known forces acting on the system. In this system, the forces acting produces torque (4.13).

$$Q_j = \tau_j \quad (4.13)$$

The torque acting on the system is the torque given by the rotors and the friction in the joints. The torque from the rotors is dependent on the distance from the rotational axis to the rotors and the angles as shown in equation (4.14) and (4.15).

$$\tau_\theta = Ku_1(t)d_{pr} \cos(pr_{tilt}) - \dot{\theta} \beta_\theta \quad (4.14)$$

$$\tau_\phi = (-Ku_2(t)d_{yr} + Ku_1(t)d_{pr} \sin(pr_{tilt})) \cos(\theta) - \dot{\phi} \beta_\phi \quad (4.15)$$

4 Dynamic Model of two Degrees of Freedom Helicopter Model

Where:

- K = Gain in rotor in newton per volt
- $u_1(t)$ = Volt applied to rotor 1, pitch rotor
- $u_2(t)$ = Volt applied to rotor 2, yaw rotor
- d_{pr} = Distance from rotational axis to pitch rotor
- d_{yr} = Distance from rotational axis to yaw rotor
- pr_{tilt} = Tilt of pitch rotor
- β_θ = Friction factor for $\dot{\theta}$
- β_ϕ = Friction factor for $\dot{\phi}$

The equations (4.7) to (4.15) are restructured to the two resulting differential equations seen in (4.16) and (4.17).

$$\ddot{\theta} = \frac{Ku_1(t)d_{pr} \cos(pr_{tilt}) - mgd_{cm} \cos(\theta) - \dot{\theta} \beta_\theta}{I_\theta} \quad (4.16)$$

$$\ddot{\phi} = \frac{(-Ku_2(t)d_{yr} + Ku_1(t)d_{pr} \sin(pr_{tilt})) \cos(\theta) - \dot{\phi} \beta_\phi}{I_\phi} \quad (4.17)$$

5 Calculating Model Parameters using the Real System

This chapter deals with the experiments and calculations needed for calculating the different parameters in the mathematical model as well as model verification.

5.1 Calculating Mass of the Arm

The mass of the arm is calculated by measuring the relative weight of the arm \tilde{m} at the center of the pitch rotor and multiplied with the distance from center of arm to rotational axis, d_{pr} , divided by distance to center of mass from rotational axis, d_{cm} , as seen in equation (5.1).

$$m = \tilde{m} \frac{d_{pr}}{d_{cm}} \quad (5.1)$$

The measurement was done with a kitchen weight and the measuring setup is shown in Figure 5-1.

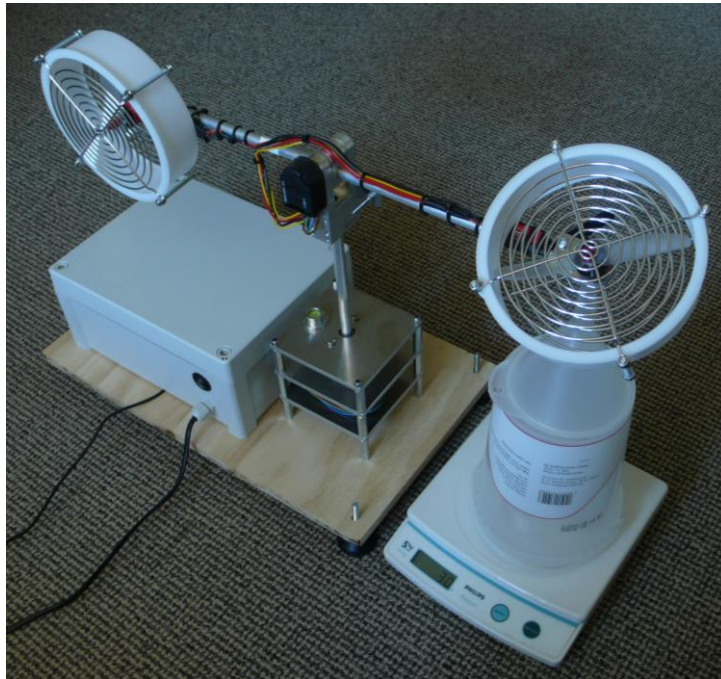


Figure 5-1: Measuring setup for mass calculation.

The mass was first calculated with a d_{cm} at 40 mm and the moment of inertia was calculated using this result. At a later stage, the distance to center of mass, d_{cm} , was changed to 15 mm so that the lower end of the rotors capacity was utilized because of the loud noise from the rotors.

5 Calculating Model Parameters using the Real System

5.1.1 Mass Calculation with Center of Mass at 40 mm

The relative mass \tilde{m} measured at the center of the pitch motor was 77 g, $d_{cm} = 41.5$ and $d_{pr} = 251$ mm. The mass of the arm is calculated to be 466 g. d_{yr} is measured to be 168 mm.

5.1.2 Mass Calculation with Center of Mass at 15 mm

The relative mass \tilde{m} measured at the center of the pitch motor was 30 g, $d_{cm} = 15$ and $d_{pr} = 226$ mm. The mass of the arm is calculated to be 452 g. d_{yr} is measured to be 196 mm.

5.1.3 Reasons for Difference in Mass Calculation

One of the reason for the change in the calculated mass of the arm is that the arm hangs below the axis of rotation. The distance measured is to the two ends and the middle is calculated. The center of mass is probably above the center of the pipe of the arm and most certainly below the rotational axis as showed by the arrows in *Figure 5-2*.

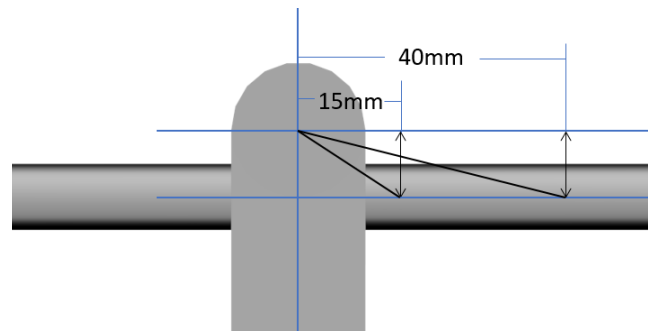


Figure 5-2: The arm of the helicopter model is hanging below the rotational axis and the exact point and distance to the of center of mass is difficult to pinpoint.

One of the results of this is that the part of the model that deals with the acceleration by gravity is less accurate the shorter the distance to the center of gravity is.

The term is kept as it is, but one could make a more accurate measurement of the distance to the center of gravity \tilde{d}_{cm} and calculate the offset angle x between the angle of the arm and the angle to center of mass, as seen in (5.2).

$$mg\tilde{d}_{cm}\cos(\theta - x) \quad (5.2)$$

5.2 Data Acquisition using Simulink

The real model is connected to the computer using a National Instrument USB DAQ using Simulink. *Figure 5-3* shows the setup in Simulink and how the data is sent to the MATLAB Workspace for further data processing.

5 Calculating Model Parameters using the Real System

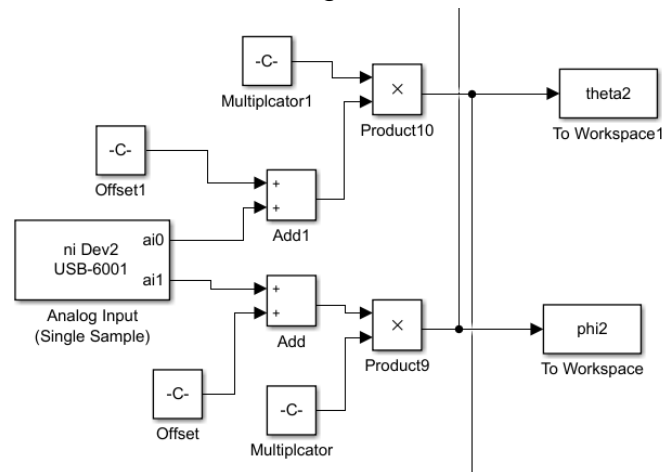


Figure 5-3: Simulink setup for Data Acquisition from real helicopter model.

The DAQ reads voltage signals and converted to radians using equations (5.3) and (5.4). The equations are created simply by taking the voltage at the zero angle and the angle and voltage at the lowest angles the helicopter model can have.

$$\theta = (V_{\theta} - 2.395) 0.354 \quad (5.3)$$

$$\phi = (V_{\phi} - 2.479) 0.639 \quad (5.4)$$

Where:

- V_{θ} = Voltage read by DAQ for angle θ
- V_{ϕ} = Voltage read by DAQ for angle ϕ

5.3 Calculating Moment of Inertia

The moment of inertia for pitch rotation is calculated by brute force optimization, meaning testing all values at given intervals in a span to find the one that fits best. Below is showed the MATLAB code used in the brute force optimization.

```
%Initialization
bestI=0;
I_range=0.005:0.0001:0.03;
error=100000;
minerror=100000;

for i1=1:length(I_range)
    error=sum(abs((m*g*d1*cos(t1theta(51:186,2))/I_range(i1))-t1theta_dd(50:185,2)));
    if error<minerror
        minerror=error;
        bestI=I_range(i1);
    end
end

end
```


5 Calculating Model Parameters using the Real System

This is done using measured data and model generated data. Equation (4.16) is used where control input 1 $u_1(t)$ is 0 and the friction factor β_θ is assumed to be so small it can be ignored. The resulting equation is shown in (5.5).

$$\ddot{\theta} = \frac{mgd_{cm}\cos(\theta)}{I_\theta} \quad (5.5)$$

The arm of the helicopter model is given a push up and caught by hand on the way down. The position data contains noise and the noise is amplified when the derivative is calculated as seen in Figure 5-4.

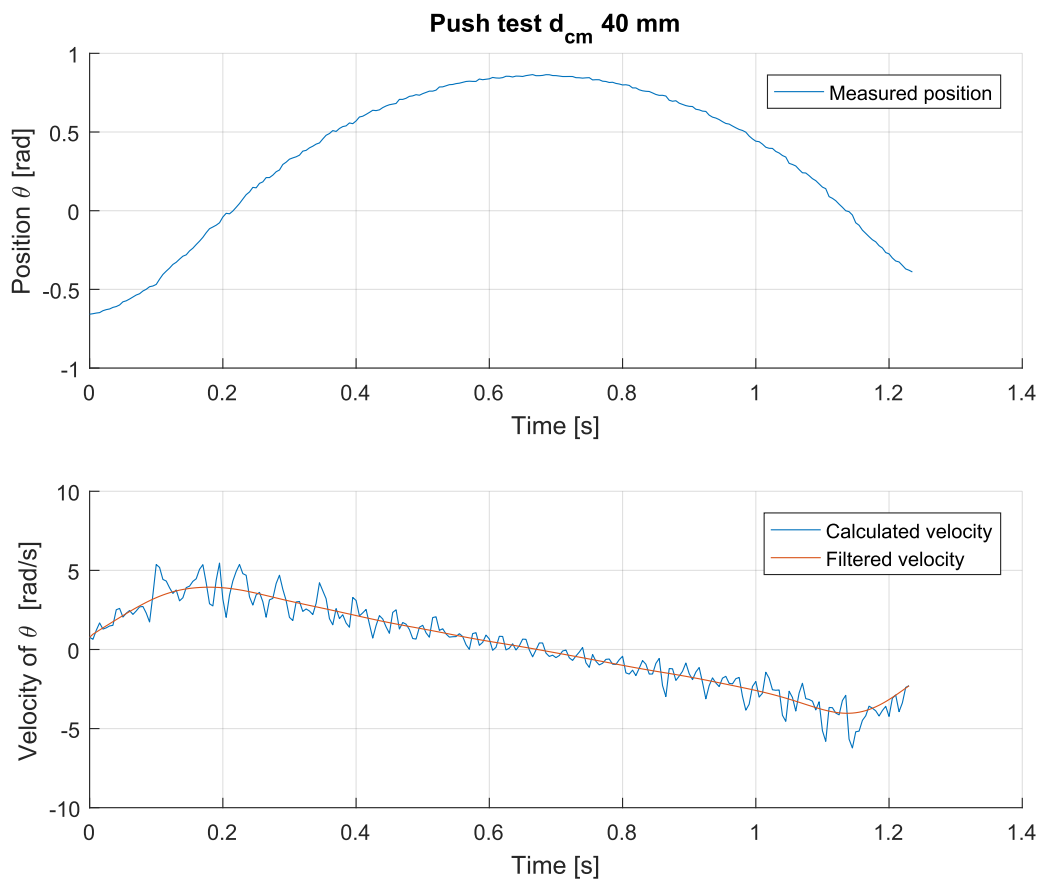


Figure 5-4: Position and velocity of the arm of the helicopter model.

The data is therefore filtered with a weighted mean filter at each stage as the noise is amplified each time the derivative is calculated.

With the filtered acceleration calculated, the unknown moment of inertia in the simplified model in (5.5) is calculated. This is done by finding the value that minimizing the difference e between the calculated acceleration $\ddot{\theta}_t$ from the push test and the acceleration given by the model $\ddot{\theta}_t$ between the time the arm leaves the hand, a , and is caught again, b , as seen in(5.6).

5 Calculating Model Parameters using the Real System

$$e = \sum_{t=a}^b |\ddot{\theta}_t - \tilde{\ddot{\theta}}_t| \quad (5.6)$$

The moment of inertia is calculated for both d_{cm} 40 and 15 mm. The moment of inertia I_{θ} was found to be 0.0170 kgm² with d_{cm} at 40 mm and 0.0159 kgm² with d_{cm} at 15 mm. Figure 5-5 and Figure 5-6 shows the acceleration calculated in the tests with the acceleration predicted by the model in the experiments conducted for finding the moment of inertia to be used in the model.

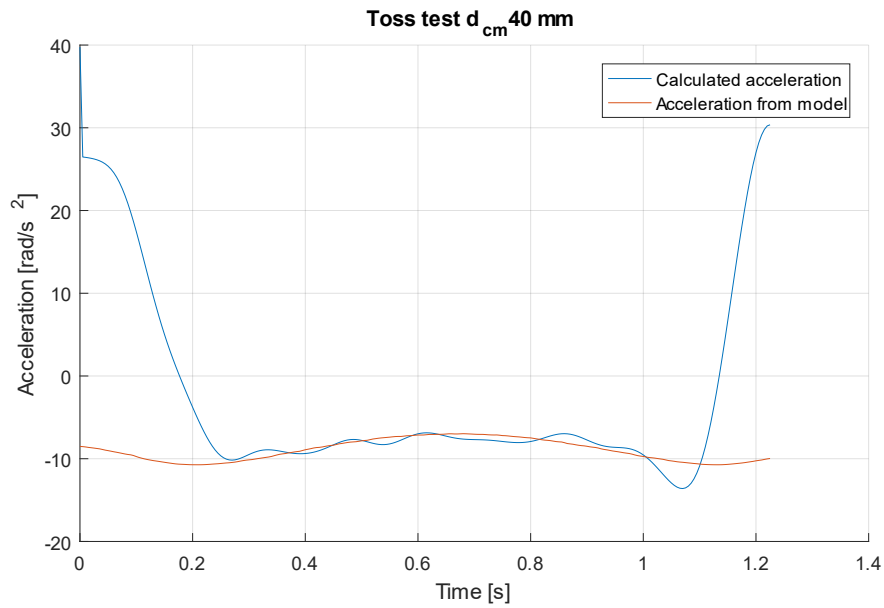


Figure 5-5: Calculated acceleration from toss test and acceleration from model with $I_{\theta} = 0.0170$ kgm² with distance to center of mas at 40 mm.

5 Calculating Model Parameters using the Real System

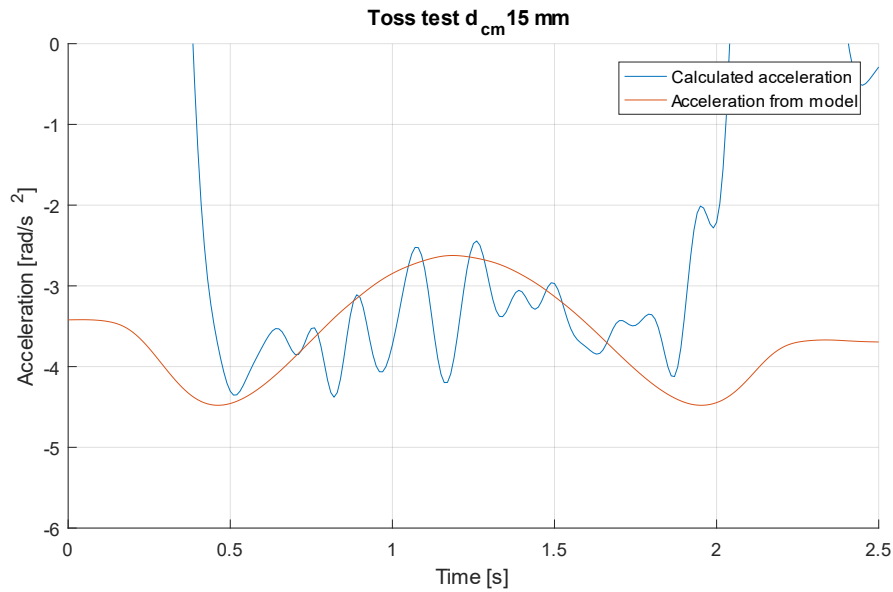


Figure 5-6: Calculated acceleration from toss test and acceleration from model with $I_0 = 0.0159 \text{ kgm}^2$ with distance to center of mass at 15 mm.

5.4 Calculating of Rotor Gain

The rotor gain is calculated by measuring the force the rotors provide at 0.5 V intervals from 0 to 5 V. The pitch rotor is turned so it faces straight down and the arm of the helicopter is horizontal so that all of the force of the rotor is straight down. The same kitchen weight used earlier and shown in *Figure 5-1* is used and the setup is the same, but with the rotor facing down. The kitchen weight gives the measurement in grams, so the measurements is multiplied by the gravity constant divided by 1000. The results are shown in *Figure 5-7*.

5 Calculating Model Parameters using the Real System

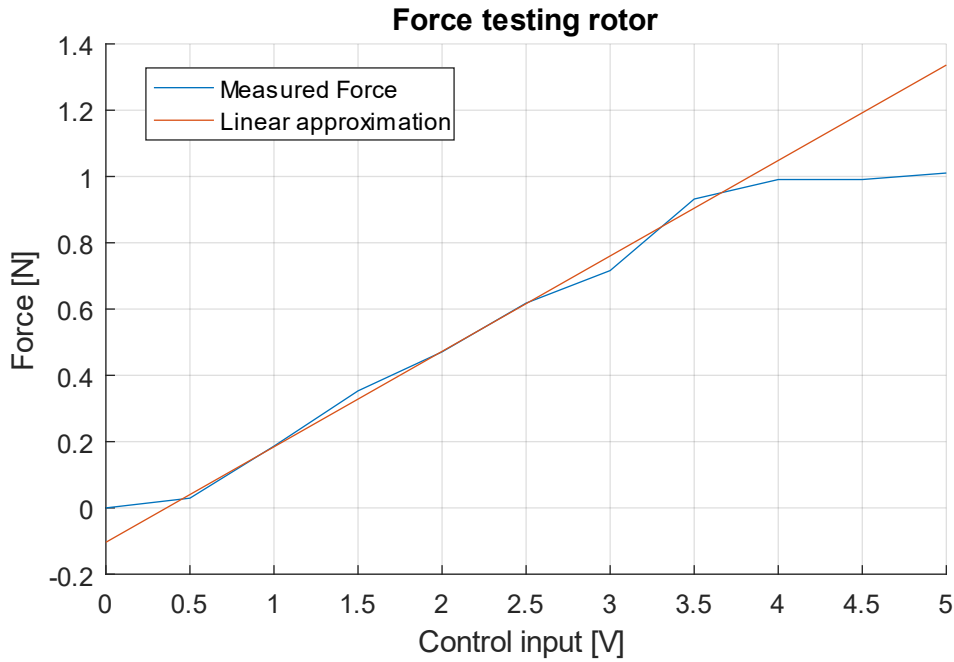


Figure 5-7: Control input to force testing of rotor. The linear approximation has a slope of 0.288 N/V, a zero offset of -0.104 N and crosses the zero line at 0.361 V.

The linear approximation is calculated with the values from 0.5 V to 3.5 V as the force given by the rotor saturates above 3.5 V and the rotor does not rotate with a control input below ca. 0.4 V. The linear approximation has a slope, rotor gain and K in equation (4.16) and (4.17), of 0.288 N/V, a zero offset of -0.104 N and crosses the zero line at 0.361 V.

Because of the dead-zone at the lower end of the control signal, the control input $u(t)$ in the mathematical model is adjusted with equation (5.7).

$$u(t) = u(t) - 0.361 \quad (5.7)$$

5.5 Calculating Friction Factor β_ϕ

When the arm of the helicopter model is horizontal, $\theta = 0$, the difference in the moment of inertia around the two axis is closest to zero. The arm with the rotors makes up all of the moment of inertia in θ axis rotation and almost all the moment of inertia in ϕ axis rotation.

The friction factor is calculated with the same method as the moment of inertia in chapter 5.2. In (5.8), difference e is minimized by finding the value for the friction factor β_ϕ that gives the least difference between the calculated acceleration from the experiment $\ddot{\phi}_t$ and the acceleration given by the simplified model $\tilde{\ddot{\phi}}_t$ seen in (5.9).

$$e = \sum_{t=a}^b |\ddot{\phi}_t - \tilde{\ddot{\phi}}_t| \quad (5.8)$$

The model (5.9) is simplified by setting all the control inputs to zero.

5 Calculating Model Parameters using the Real System

$$\ddot{\phi} = \frac{\dot{\phi} \beta_{\phi}}{I_{\phi}} \quad (5.9)$$

The friction factor, β_{ϕ} , is calculated between $a = 0.7$ and $b = 2.6$ s in the plot showed in *Figure 5-8* and was calculated to be 0.02.

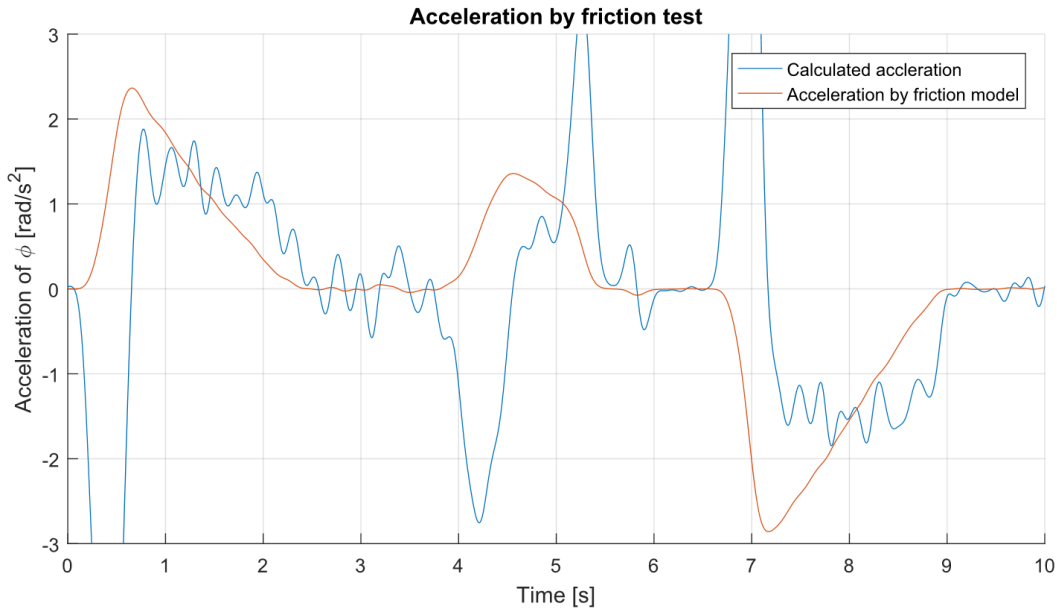


Figure 5-8: Acceleration caused by friction. The arm is kept horizontal and given a push and let go until it stops from friction. In one direction from 0.5 to 3 seconds and back again from 7 to 9 seconds.

Looking at the calculated acceleration in *Figure 5-8* it appears that a friction model based only on the direction of movement would fit the data well. To test this a modified model of the acceleration by friction was tested and seen in equation (5.10).

$$\ddot{\phi} = \frac{\text{sign}(\dot{\phi})\beta_{\phi 2}}{I_{\phi}} \quad (5.10)$$

The friction factor $\beta_{\phi 2}$ is calculated with the same method and same data as earlier and found to be 1.1520 and the resulting plot is shown in *Figure 5-9*.

5 Calculating Model Parameters using the Real System

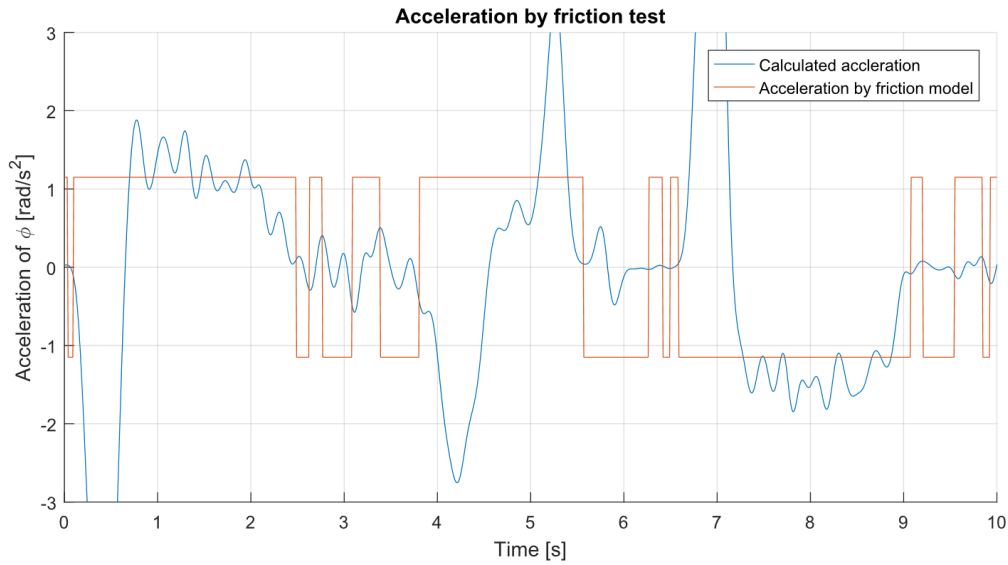


Figure 5-9: Acceleration by friction with friction model found in equation (5.10).

Although the friction model where the friction is only dependent on the direction of the velocity, (5.10), fits well, and one might argue better in at least one direction than the velocity dependent model, the velocity dependent model in (5.9) is chosen as this can be linearized. A combination was also thought to work well but was discarded for the same reason.

5.5.1 Finding Friction Factor β_θ

The friction factor β_θ was not calculated but found during later experiments, through trial and error, to be half of β_ϕ giving β_θ a value of 0.01.

5.6 Building the Model in Simulink

The Simulink model is built using built-in blocks and the complete model with the inputs and outputs to the real helicopter model is shown in Figure 5-10.

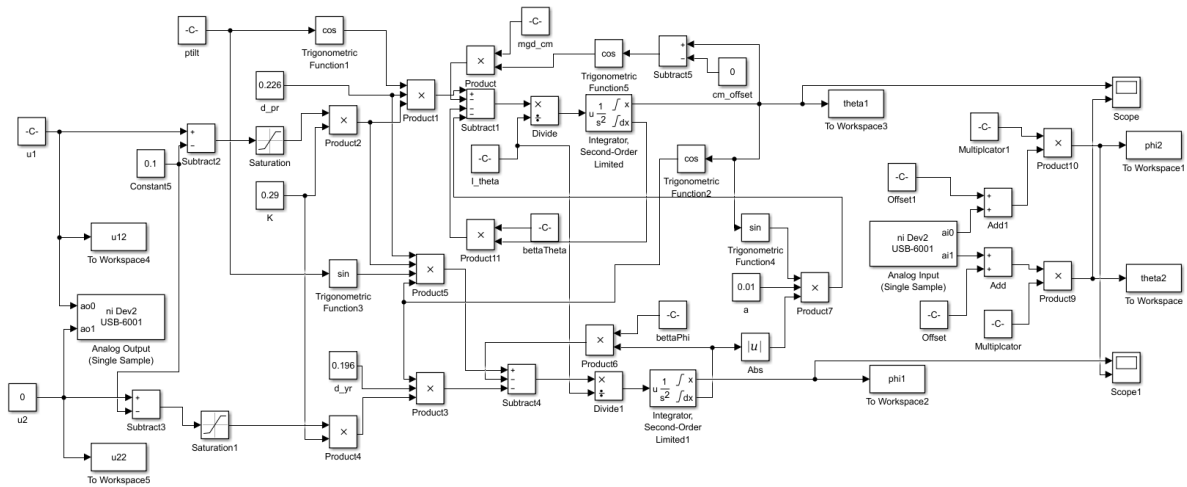


Figure 5-10: Complete mathematical model with DAQ inputs and outputs to real helicopter model.

5 Calculating Model Parameters using the Real System

Not shown in *Figure 5-10* are slider controls, connected to block u1 and u2, to set the control inputs.

To fit the mathematical model to the real helicopter model, the rotor dead-zone, calculated to be 0.36 and seen in equation (5.7) is reduced through trial and error to 0.1. It is named Constant5 in the Simulink implementation shown in *Figure 5-10*.

As discussed in chapter 5.1.3, another approach to adjust the fit of the mathematical model to the real model is finding the offset angle between the angle of the arm and the angle to center of mass. This option was not utilized but the option was implemented in the Simulink model and a cutout of the model is shown in *Figure 5-11* with the option in the red rectangle.

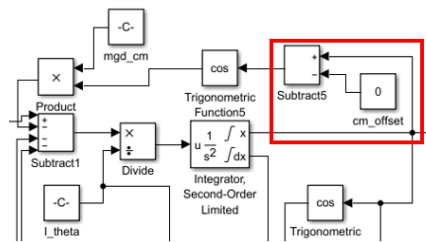


Figure 5-11: Offset angle between the angle of the arm and angle to center of mass was implemented but not utilized.

During testing of the mathematical model against the real helicopter model, the arm of the helicopter was lifted during quick yaw action. This was attributed to the centripetal force and a new term entered the model. Equation (4.16) was updated to (5.11) and a cutout of the Simulink model is shown in *Figure 5-12* with the new term in the red rectangle.

$$\ddot{\theta} = \frac{Ku_1(t)d_{pr} \cos(pr_{tilt}) - mgd_{cm} \cos(\theta) - \dot{\theta} \beta_{\theta} - \alpha |\dot{\phi}| \sin(\theta)}{I_{\theta}} \quad (5.11)$$

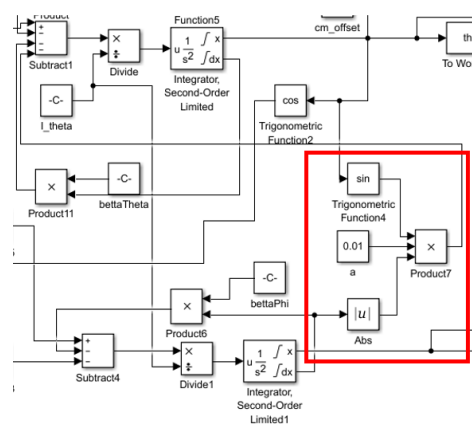


Figure 5-12: New term to the model to handle the result of the centripetal force.

The new term introduces a new constant, α , and this was determined to be 0.01 through trial and error.

Though the new term is relevant in fast yaw action, that is not something that is happening most of the time; and the term can be neglected for low speed applications. The absolute value of the speed was also used but later research show that the squared speed is commonly used.

5 Calculating Model Parameters using the Real System

5.7 Model Tuning

With all the parameters found, the mathematical model was fitted to the real model by adjusting through trial and error the dead-zone in the rotors from the found 0.361 to 0.1. The result is shown in Figure 5-13. As this simple adjustment worked so well, no other parameters were changed. Choosing this adjustment means that the model works best in the lower part of the available pitch range.

The figure shows the position in both radians and degrees for easier visualization, as most people are most used to degrees as the unit for the angle.

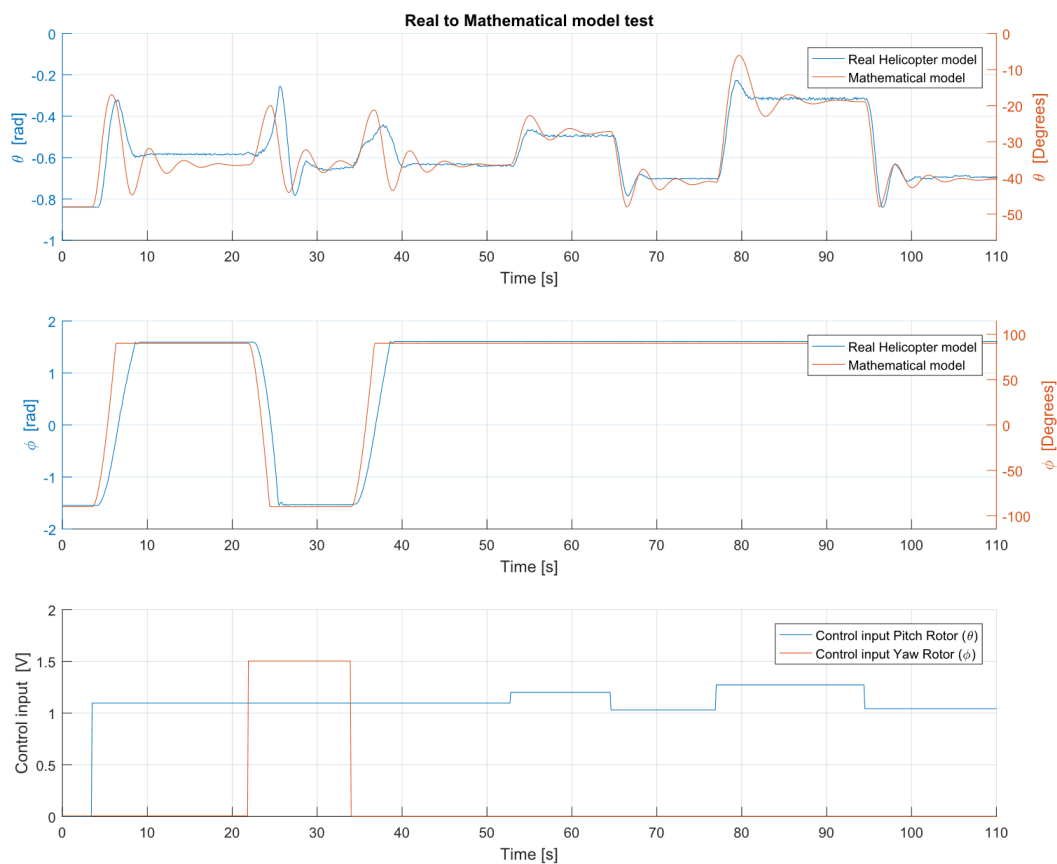


Figure 5-13: Mathematical model tested against real helicopter model.

The reason for the apparent time delay at the start and in the second plot is because here the change in the control signal is so large that the rotors need time to speed up. Later in the plot the time delay is negligible when the change in the control signal is much smaller.

Movement in position ϕ from one side to the other shows that the model fits better for one direction than the other. The movement from 90° to -90° , about 24 seconds into the test, shows a very good fit for the model, but movement in the other direction does not fit as well.

5 Calculating Model Parameters using the Real System

For movement in position θ , the model does a small overshoot but has a good fit for the steady state value.

Because the arm of the helicopter is forced against the end-guards during almost the entire time in the plot showed above, it appears that the friction for pitch movement is wrong, but part of this is a result of the arm pushed at the end-guards and is much higher than for movement free of the end-guards.

5.8 Complete Model with all Parameters Quantified

This chapter is included for easy access to the complete model (5.12) and (5.13), with all parameters quantified as used in the Simulink model showed in chapter 5.6.

$$\ddot{\theta} = \frac{Ku_1(t)d_{pr} \cos(pr_{tilt}) - mgd_{cm} \cos(\theta) - \dot{\theta} \beta_{\theta} - \alpha |\dot{\phi}| \sin(\theta)}{I_{\theta}} \quad (5.12)$$

$$\ddot{\phi} = \frac{(-Ku_2(t)d_{yr} + Ku_1(t)d_{pr} \sin(pr_{tilt})) \cos(\theta) - \dot{\phi} \beta_{\phi}}{I_{\phi}} \quad (5.13)$$

Where:

- θ - Pitch position
- ϕ - Yaw position
- $K=0.288$ - Gain in rotor in newton per volt
- $u_1(t) = u_1(t) - 0.1$ - Volt applied to rotor 1, pitch rotor
- $u_2(t) = u_2(t) - 0.1$ - Volt applied to rotor 2, yaw rotor
- $d_{cm} = 15$ mm - Distance from rotational axis center of mass
- $d_{pr} = 226$ mm - Distance from rotational axis to pitch rotor
- $d_{yr} = 196$ mm - Distance from rotational axis to yaw rotor
- $pr_{tilt} = 35^\circ$ - Tilt of pitch rotor
- $\beta_{\theta} = 0.01$ - Friction factor for $\dot{\theta}$
- $\beta_{\phi} = 0.02$ - Friction factor for $\dot{\phi}$
- $g = 9.81$ - Gravity constant
- $m = 0.452$ kg - Mass of arm
- $\alpha = 0.01$ - Constant factor for action by centripetal force

6 PID Control

This chapter deals with building the PID controllers in Simulink and finding PID parameters.

6.1 PID Controller on Velocity Form

Starting with the PID equation on the positional form as seen in (6.1), the derivative is taken and one get the PID algorithm on velocity form seen in equation (6.2).

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_{-\infty}^t e(\tau) d\tau + K_p T_d \frac{de(t)}{dt} \quad (6.1)$$

$$\frac{d}{dt} u(t) = K_p \frac{de(t)}{dt} + \frac{K_p}{T_i} e(t) + K_p T_d \frac{d^2 e(t)}{dt^2} \quad (6.2)$$

As the velocity form gives the change of the control input it is very simple to set limits on the rate of change in the control signal and this is used in this application. This is done to reduce the impact the noise can have on the control.

The algorithm is implemented in Simulink as seen in *Figure 6-1*, where “Saturation2” block is the one limiting the change in the control signal u_1 . The change integrated and one gets the control signal.

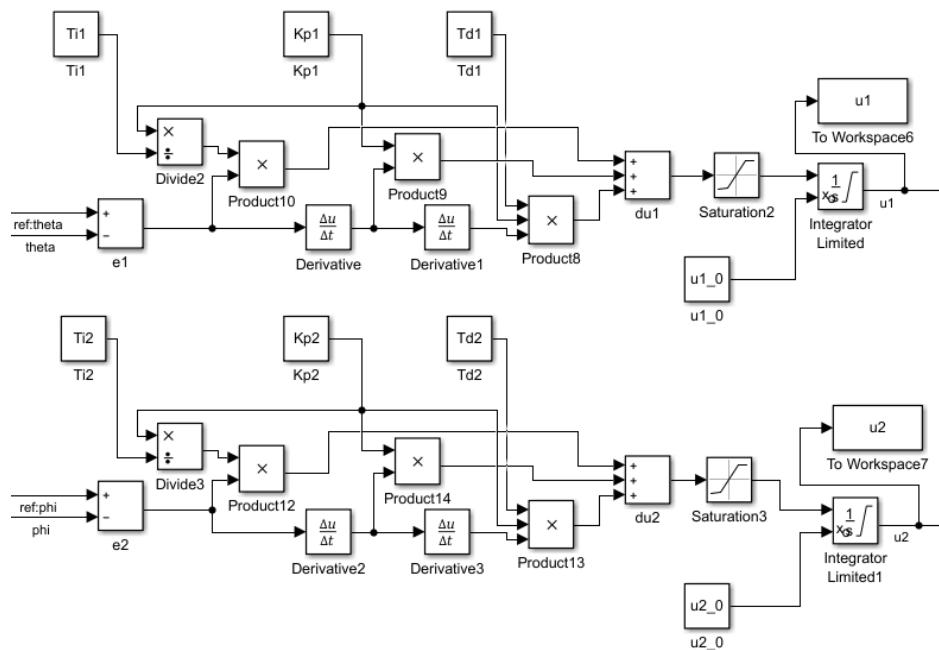


Figure 6-1: Simulink implementation of the PID controllers on velocity form.

6.2 PID Parameters

There are multiple methods for finding good PID parameters with Skogestad model based and Ziegler–Nichols closed loop methods as the most commonly known. Having a non-linear model in Simulink of the process, good PID parameters can be found using optimization. Although not commonly used, this has been done with great success on various processes during the master program.

Brute force optimization was used in the beginning, as this method ensures that one finds the global minimum if it is within the range of the values one is testing. Not knowing in what the range the good values for the PID parameters are meant that a sizable range with high precision was necessary to go through. Testing every parameter combination for the PID parameters from 0 to 3 at 0.01 precision meant that $300^3 = 2\,700\,000$ tests were needed to be made for each PID. This was not feasible, with each test taking a little over one second.

Therefore, a numerical nonlinear solver called `fmincon()` was used in MATLAB. This function does not guarantee that a global minimum has been found. The function to minimize is shown in equation (6.3).

$$e = \sum_{t=0}^{end} \sqrt{|\theta_r - \theta|} + \sqrt{|\phi_r - \phi|} \quad (6.3)$$

The root of the absolute difference between the reference and the simulated value was chosen as the function, because previous experience has showed that a lower exponential gives a less aggressive control action.

The testing is done on the combined mathematical model, seen in *Figure 5-10*, and PID implementation, seen in *Figure 6-1*, in Simulink where noise has been added to the position signal with a lowpass filter with time-constant of 0.2 s.

The friction factor β_θ is adjusted to a lower value to make the control more difficult for the PID controller to reflect the random oscillations found in the real system. Different variances in the added noise was also tested to get multiple PID parameters to test on the real system.

After `fmincon()` has found the optimal values for the PID controllers, the test are done once more and the result of the simulation is plotted for a visual check of the result of the optimal PID parameters. One of these plots is shown in *Figure 6-2*, where the PID parameter values for a timestep at 0.02 s was found.

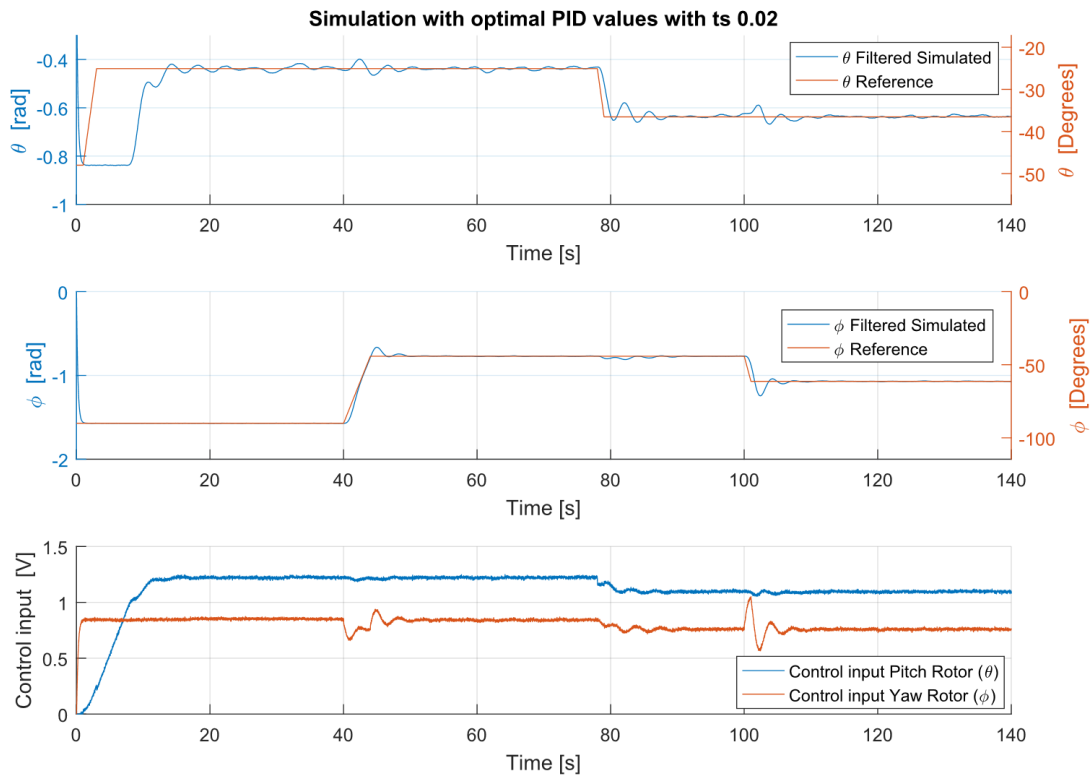


Figure 6-2: Plot of the PID with optimal parameters used on the nonlinear model.

First PID parameters was found for a timestep of 0.1 seconds, these was tested and the best result is shown in Figure 6-3. The PID parameters used here are $K_{p\theta} = 0.623$, $T_{i\theta} = 0.735$, $T_{d\theta} = 0.499$, $K_{p\phi} = -2.997$, $T_{i\phi} = 1.772$ and $T_{d\phi} = 0.125$.

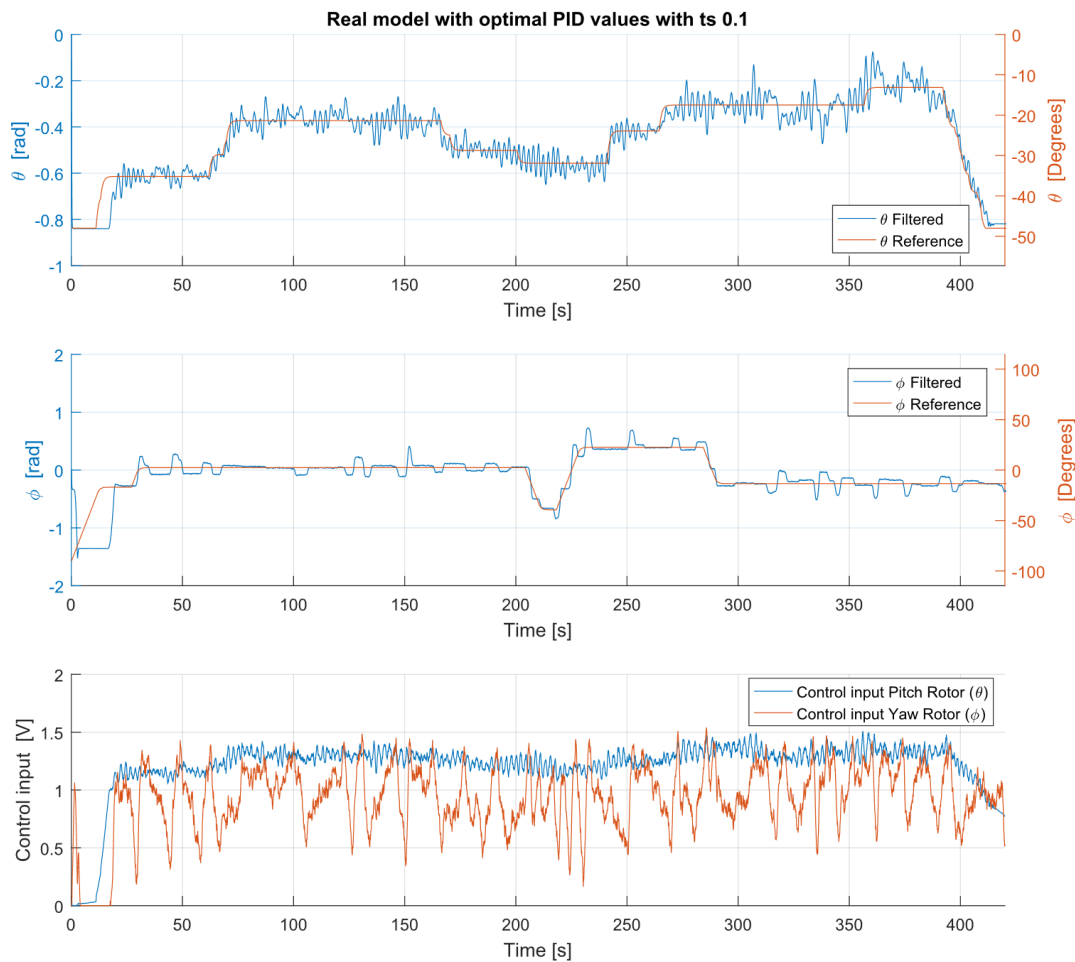


Figure 6-3: PID control of real model using optimal PID parameter found for timestep of 0.1 seconds.

The control is not very good with θ oscillating rapidly with an maximum amplitude of 10° , ϕ never gets to the setpoint but overshoots in both directions. Because of this, it was decided to reduce the timestep to 0.01 seconds. New PID parameters were found, but the real-time pacer that keeps the calculations in real time, showed that the Simulink could not get all the calculations, setting and reading values from the DAQ done on time all the time. Figure 6-4 shows the elapsed time in Simulink minus time elapsed in the real world is not kept close to constant all the time.

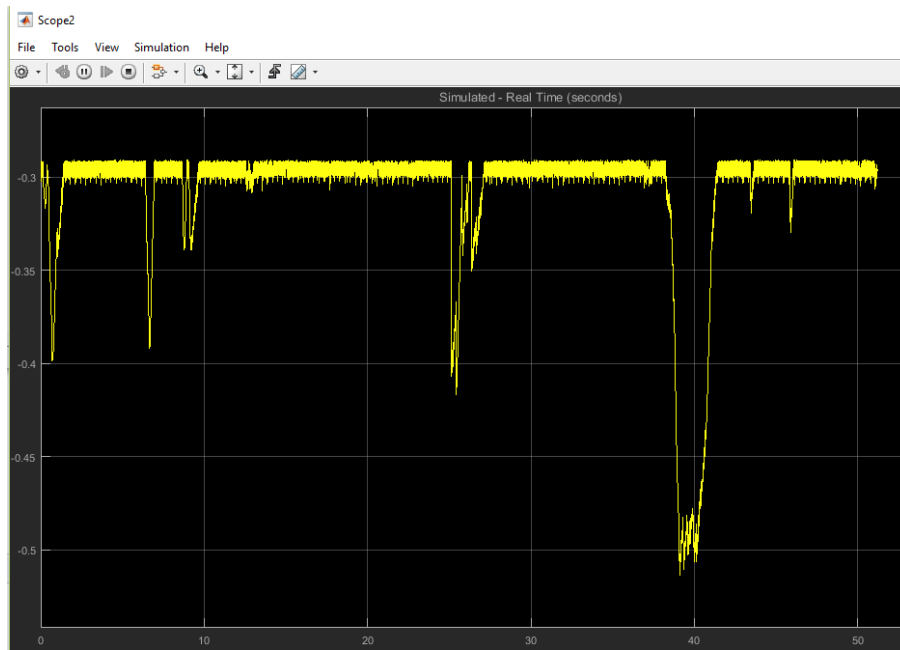


Figure 6-4: Plot of the time in elapsed Simulink minus elapsed time in the real world.

The timestep was increased to 0.02 and the Simulink solver was set to ode1(Euler). Simulink was able to keep up with real time and new parameters for the PID was found. The results of the parameters that was found to best control the helicopter model is shown in Figure 6-5. The PID parameters used here is $K_{p\theta} = 0.190$, $T_{i\theta} = 0.477$, $T_{d\theta} = 1.851$, $K_{p\phi} = -1.440$, $T_{i\phi} = 2.58$ and $T_{d\phi} = 0.231$.

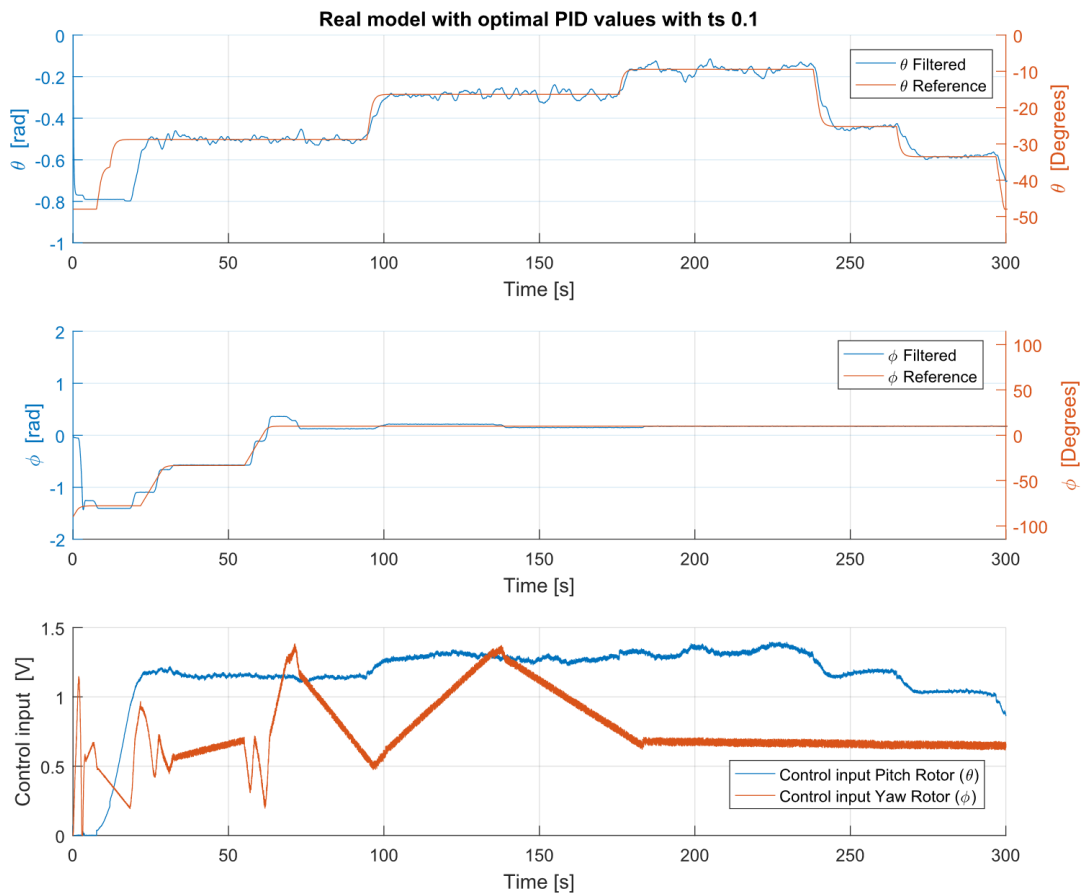


Figure 6-5: PID control of real model using optimal PID parameter found for timestep of 0.02 seconds.

This control can be said to be much better, ϕ gets close to the setpoint relative quick and each time it moves it gets closer to the setpoint. θ is not kept at the set point but moves at random about 3° above and below the setpoint. The control signal line looks to be very thick, but this is just the noise in the measured signal bleeding through to the control signal.

One factor neglected throughout the study, the static friction in yaw joint, is very easy to see by looking at these plots. The PID controller overcomes the friction slowly and gets the yaw angle to setpoint.

7 MPC Control

This chapter covers the steps needed in development of a MPC controller for the

7.1 Linear Model

In order to perform linear MPC control a linear model of the system is needed. The linear model is also needed for the Kalman filter that is to be used for state estimation. To achieve this a Jacobian linearization is performed around a steady state working point. The working point selected is:

- $\theta_0 = -0.27 = -15.5^\circ$
- $\dot{\theta}_0 = 0$
- $\phi_0 = 0$
- $\dot{\phi}_0 = 0$
- $u_1^0 = 1.28$
- $u_2^0 = 0.90$

7.1.1 Linearization

This chapter covers a simplified description of how the Jacobian linearization is performed.

The model developed is non-linear, and can be expressed on the form seen in (7.1) and (7.2).

$$\dot{x} = f(x, u) \quad (7.1)$$

$$y = g(x, u) \quad (7.2)$$

A Taylor series expansion is performed where all higher than 1st order terms are ignored on the right hand side of equations (7.1) and (7.2) around the selected working point and we get equations (7.3) and (7.4).

$$f(x, u) \approx f(x_0, u_0) + \left. \frac{df}{dx^T} \right|_{x_0 u_0} (x - x_0) + \left. \frac{df}{du^T} \right|_{x_0 u_0} (u - u_0) \quad (7.3)$$

$$g(x, u) \approx g(x_0, u_0) + \left. \frac{dg}{dx^T} \right|_{x_0 u_0} (x - x_0) + \left. \frac{dg}{du^T} \right|_{x_0 u_0} (u - u_0) \quad (7.4)$$

The deviation variables defined as

$$\Delta x = x - x_0 \quad (7.5)$$

$$\Delta u = u - u_0 \quad (7.6)$$

As we have a steady state point where the linearization is performed around the constant term in (7.3) is zero, as seen in (7.7).

$$f(x_0, u_0) = 0 \quad (7.7)$$

Matrix A, B, C and D is defined as

$$A = \left. \frac{df}{dx^T} \right|_{x_0 u_0}, \quad B = \left. \frac{df}{du^T} \right|_{x_0 u_0} \quad (7.8)$$

$$C = \left. \frac{dg}{dx^T} \right|_{x_0 u_0}, \quad D = \left. \frac{dg}{du^T} \right|_{x_0 u_0}$$

And so we have a linearized state model as seen in (7.9).

$$\begin{aligned} \dot{\Delta x} &= A\Delta x + B\Delta u \\ \Delta y &= C\Delta x + D\Delta u \end{aligned} \quad (7.9)$$

The constant term in (7.4) is y_0 as seen in (7.10) and the deviation variable Δy is defined in (7.11).

$$g(x_0, u_0) = y_0 \quad (7.10)$$

$$\Delta y = y - y_0 \quad (7.11)$$

Last the deviation variables is redefining as seen in (7.12) and (7.13).

$$x = x - x_0 \quad (7.12)$$

$$u = u - u_0 \quad (7.13)$$

7.1.2 Linearized Model

First the states and outputs is defined:

- $x_1 = \theta$
- $x_2 = \dot{\theta}$
- $x_3 = \phi$
- $x_4 = \dot{\phi}$
- $y_1 = x_1$
- $y_2 = x_3$

There are four states and we have four equations for the change in the states seen in (7.14) to (7.17).

$$\dot{x}_1 = x_2 \quad (7.14)$$

$$\dot{x}_2 = \frac{Ku_1(t)d_{pr} \cos(pr_{tilt}) - mgd_{cm} \cos(\theta) - \dot{\theta} \beta_\theta - \alpha |\dot{\phi}| \sin(\theta)}{I_\theta} \quad (7.15)$$

$$\dot{x}_3 = x_4 \quad (7.16)$$

$$\dot{x}_4 = \frac{(-Ku_2(t)d_{yr} + Ku_1(t)d_{pr} \sin(pr_{tilt})) \cos(\theta) - \dot{\phi} \beta_\phi}{I_\phi} \quad (7.17)$$

Using (7.8) on these equations we get the following linearized model matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{mgd_{cm} \sin(\theta_0)}{I_\theta} & -\frac{\beta_\theta}{I_\theta} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\beta_\phi}{I_\phi} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ \frac{Kd_{pr} \cos(pr_{tilt})}{I_\theta} & 0 \\ 0 & 0 \\ \frac{Kd_{pr} \sin(pr_{tilt}) \cos(\theta_0)}{I_\phi} & \frac{-Kd_{yr} \cos(\theta_0)}{I_\phi} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = 0$$

There are terms that have been discarded where the partial derivative did not evaluate to zero because of difference between the real system and the mathematical model.

7.2 Kalman Filter

The MPC controller needs to know all the states in the system so a state estimator is needed, the Kalman filter fulfill this role. Figure 7-1 shows how the Kalman filter is in parallel with the process estimating the states

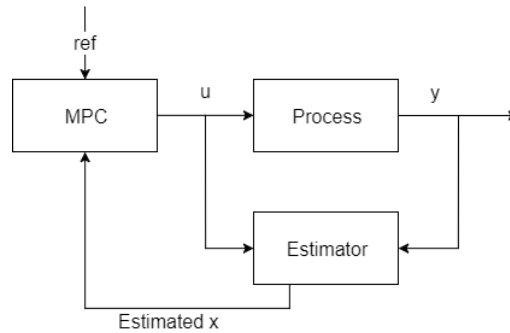


Figure 7-1: The Kalman is in parallel with the process and estimates the states.

The Kalman filter is tuned with the covariance matrices V and W and needs the matrices A , B , C and D in discretized form. The state space matrices found is discretized with the MATLAB function `c2d()`.

Tuning a Kalman filter is not trivial and after the trial and error method failed, the Kalman filter was tuned by optimization. Simulated data was recorded and the MATLAB function `fmincon()` was used. (7.18) shows the function that was minimized, the absolute of the simulated values minus the value given by the Kalman filter.

$$e = \sum_{t=0}^{end} |x_{1s} - x_{1K}| + |x_{2s} - x_{2K}| \cdot 10 + |x_{3s} - x_{3K}| + |x_{4s} - x_{41K}| \quad (7.18)$$

The optimizer found the covariance matrices be:

$$V = \begin{bmatrix} 0.000874 & 0 \\ 0 & -0.0139 \end{bmatrix}$$

$$W = \begin{bmatrix} -0.000393 & 0 & 0 & 0 \\ 0 & 0.00576 & 0 & 0 \\ 0 & 0 & 0.00175 & 0 \\ 0 & 0 & 0 & -0.009910 \end{bmatrix}$$

A new simulation was conducted and the with the Kalman filter and the Kalman filter estimates the unknown states very well but has some constant offset in the position states. This clearly seen in Figure 7-2.

Kalman filter test

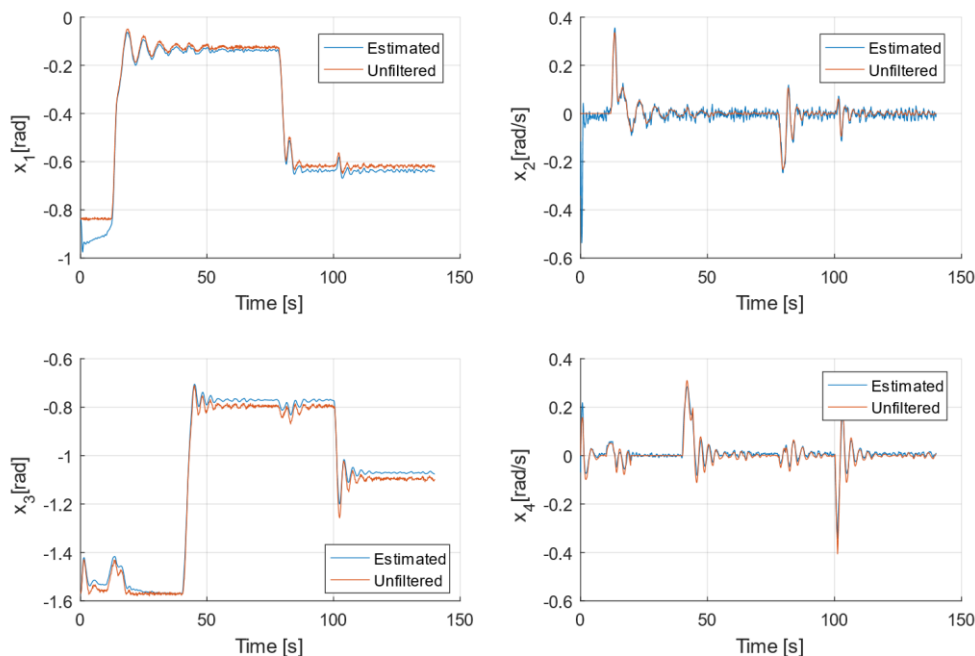


Figure 7-2: Testing the Kalman filter on new simulated data with constant offset error in position.

The offset is removed by simply taking the average difference and removing it at the end of the Kalman function. With the offset removed, it is clear that the Kalman filter filters the position data well, as can be seen in Figure 7-3.

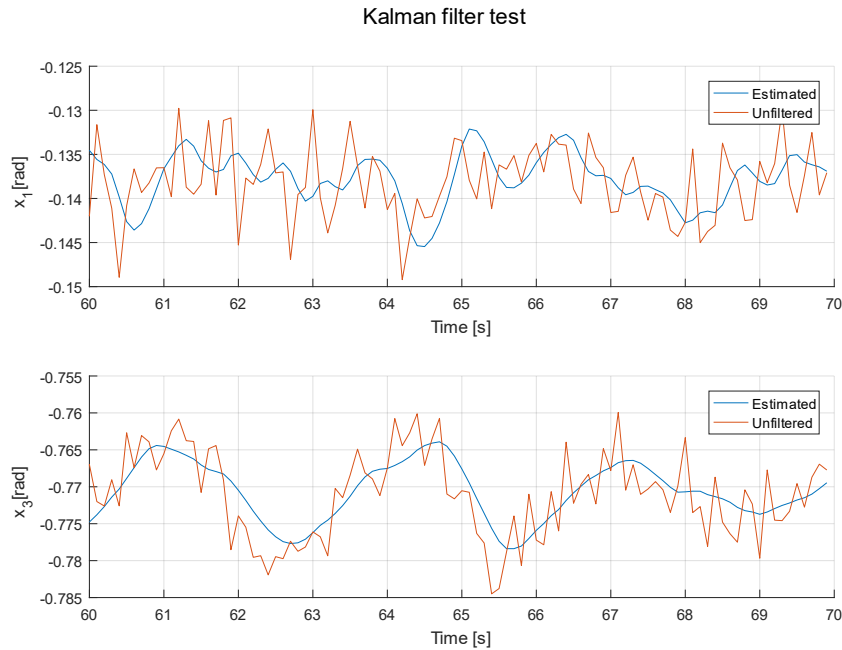


Figure 7-3: Testing the Kalman filter again after constant offset is removed.

The Kalman filter is implemented in Simulink in a MATLAB Function as seen in Figure 7-4 and the code inside the function below. W , V , C , B , A and Id is retrieved from MATLAB workspace.

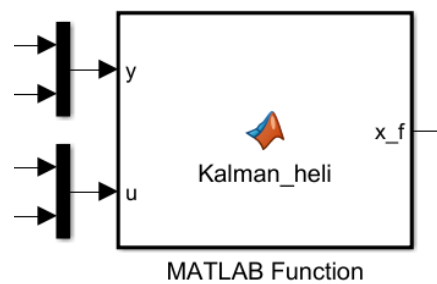


Figure 7-4: The Kalman filter implementation in Simulink.

```
function x_f = Kalman_heli(y, u, W, V, C, B, A, Id)

y=y-[-0.27;0];%delta y
u=u-[1.28;0.9];%delta u
global Pp;%Needs to be initialized as a simulink.signal
global x_hatp;%Needs to be initialized as a simulink.signal

Pm=A*Pp*A'+W;
```

```

Kk=Pm*C'*inv(C*Pm*C'+V);
x_hatm=A*x_hatp+B*u;
y_hatm=C*x_hatm;
x_hatp=x_hatm+Kk*(y-y_hatm);
Pp=(Id-Kk*C)*Pm*(Id-Kk*C)'+Kk*V*Kk';

x_f=x_hatp-[0.27-0.0124; 0; +0.0241; 0];%Back to nonlinearized values + some minor offset
adjustments

```

7.3 LQ Optimal Control

The development of a MPC controller was started, but unforeseen problems with Simulink as well as lack of time results in the unfortunate consequence that no MPC controller has been developed for the helicopter model. The solver, `quadprog()`, used in the MPC controller, is not supported for standalone code generation and the MATLAB function with the MPC controller will therefore not compile in Simulink.

8 Conclusion

A mathematical model was developed using Euler-Lagrange model formulation. By conducting experiments on the real system, most of the parameters in the model were calculated, and the rest was estimated by trial and error. The resulting model described the physical system very well, apart from the static friction which was neglected.

The model was used to develop PID controllers for the system. The PID controllers performed as well as could be expected, considering the highly unstable system and the high static friction in the yaw joint. The PID controller got the yaw angle close to the setpoint quickly, but use some time to get on the setpoint because of the static friction. The PID controller was able to keep the pitch angle close to setpoint with intermittent movement about 3° off setpoint because of random disturbances.

The model was linearized and a Kalman filter was developed for the MPC controller. The Kalman filter was able to estimate the unknown states very well and performed good filtering of the measured states.

Due to unforeseen problems with Simulink as well as lack of time results in the unfortunate consequence that no MPC controller has been developed for the helicopter model.

References

- [1] Juan Andres Gutiérrez, Jaime Arcos-Legarda " CONTROL SYSTEM FOR A TWO DEGREES OF FREEDOM HELICOPTER", presented at Conference: III Congreso Internacional de Ingeniería Mecatrónica y Automatización, CIIMA 2014
- [2] A.P.S. Ramalakshmi, P.S. Manoharan, K. Harshath, and M. Varatharajan, " Model predictive control of 2DOF helicopter ", International Journal of Innovation and Scientific Research, Vol. 24, No. 2, pp. 337-346, Jun. 2016.

Appendices

Appendix A Task Description

FMH606 Master's Thesis

Title: Modelling and control of two degrees of freedom helicopter model

HSN supervisor: Roshan Sharma, Assoc. Prof., University College of Southeast Norway

External partner: -

Task background:

At HSN, a two degree of freedom helicopter (2 DOF) model has been built from the scratch (see Figure 1). This will serve as a laboratory component in various courses at HSN in the future if successful.

A two degree of freedom helicopter model is a MIMO system with two control inputs and two control outputs. The two outputs are the pitch angle (nose up-down angle) and the yaw angle (clockwise/anti-clockwise rotational movement). Both outputs are directly measured using suitable angle sensors. The control inputs are the voltages to the lift/pitch motor and the vertical stabilizer/yaw motor.

The helicopter model mimics a real world helicopter in terms of flight dynamics. There is a strong coupling between the two control inputs and the two outputs, which makes this lab model a very good candidate for model based control.

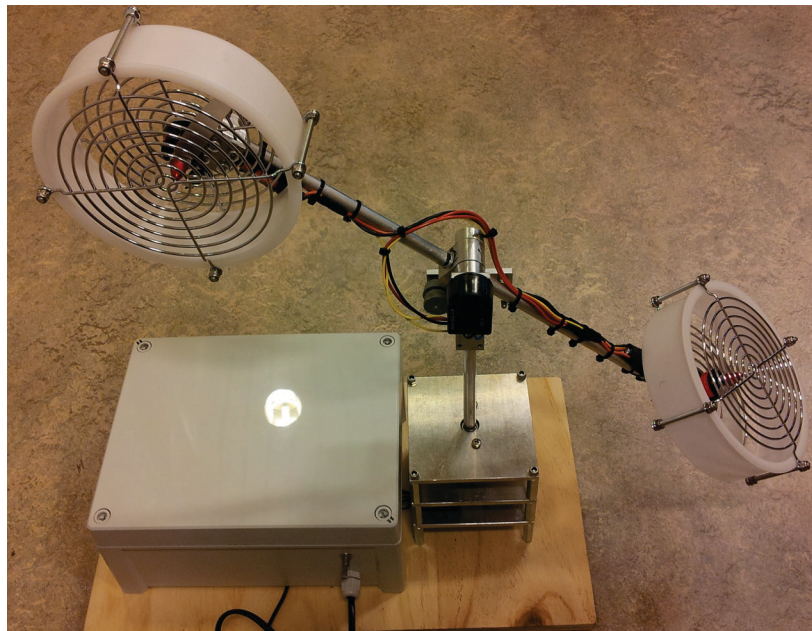


Figure 1: 2 DOF helicopter model at HSN

Position control of unstable system finds its application from aerospace industry (position control of aeroplanes, helicopter etc.) to marine industry (position control of ship, submarine etc.). The controllers can be classical PID controller to advanced model based control. For model based control, a simplified model that represents the dynamics of the system is being studied at HSN. The model will further be used to develop model based

control like MPC, LQ control etc. The student working with this thesis will have a good opportunity to contribute in model development and in advanced controller design. If successful, the students' work (model + controller) will be acknowledged and used as laboratory work in the future.

The work is suitable to students from IIA (SCE, systems and control engineering) only.

Task description:

Development of a mathematical model of a 2 DOF helicopter model and design of controllers to control the position of the helicopter are the main tasks.

1. Give an overview of the helicopter dynamics and explain how it works.
2. Perform a thorough literature survey about different kinds of models that have been used for modelling flight dynamics of 2 DOF helicopter models.
3. Develop a dynamic model of the 2 DOF helicopter model suitable for control purpose.
4. Perform experiments on the real 2 DOF helicopter model and use the experimental data for model validation and parameter estimation.
5. Develop a PID controller to control the position of the helicopter model. Test the controller with the real system.
6. Use the model of the system to develop a model based controller (for e.g. MPC or LQ controller) to control the position of the helicopter. If time permits, test the controller with the real system and compare its performance with the PID controller.
7. Report the work in the Master's Thesis. Present the thesis work.

Student category: Only for IIA / SCE students.

Practical arrangements: Experiments in the 2 DOF helicopter model at HSN.

Signatures:

Student (date and signature):

Supervisor (date and signature):