Modeling and Simulation of Stabilizer for Remote Controlled Helicopter

Jan Palmer Kolberg

Supervisors
Professor Hamid Reza Karimi
Trond Friisø

This Master’s Thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2012
Faculty of Technology
Department of Engineering
Abstract

When working with a complex system it will always be useful to try to control the systems behavior to make it follow a given command. To make this possible different approaches can be taken and solutions purposed adjusted for the given system.

The main focus of this thesis has been to look at different methods and ways to control the movement and behaviour of a radio controlled (RC) helicopter. A comprehensive litterature study is performed to present the possibilities and the different approaches to such a difficult and complex problem.

A method on how to obtain a linearized helicopter model is presented. Different control methods for multiple-input multiple-output systems is explained and tested on a linearized model. A controller for position reference is modeled with Linear quadratic regulator (LQR) with integral action in addition to a kalman filter.

This project shows how to model helicopter behavior and how this can be represented with different linearized models. A Linear quadratic gaussian controller with integral action is simulated and is shown to have satisfactory results. How to implement and test such a solution on a physical helicopter is discussed.

Keywords: Small scaled helicopter, Linearization, Integral action, Matlab/Simulink.
Acknowledgements

This thesis is carried out as a conclusion of the masters degree in mechatronics at the University of Agder. The specification for the task is given by Origo Solutions.

The author of this thesis would like to thank Professor Karimi for good support along the way. I also would like to thank friends and classmates for the assistance and continuous support throughout the project,
# Contents

1 Introduction ........................................... 5

2 Concept of Helicopter movement ..................... 6
   2.1 Introduction ........................................ 6
   2.2 Movement related to the Z axis .................... 7
      2.2.1 Cyclic Pitch .................................... 8
   2.3 Movement related to X axis ....................... 9
   2.4 Movement related to Y axis ....................... 9
      2.4.1 Summary ....................................... 10

3 Modeling of the helicopter ......................... 11
   3.1 Introduction ........................................ 11
   3.2 Nonlinear model .................................... 12
      3.2.1 Reference geometry ............................ 12
      3.2.2 Translational movement ....................... 13
      3.2.3 Angular movement .............................. 15
   3.3 State space ......................................... 17
   3.4 Linearization ....................................... 18
      3.4.1 Linearized for RC helicopter ................ 20
   3.5 Simulation model .................................. 21

4 Helicopter Control .................................. 22
   4.1 Introduction ........................................ 22
   4.2 Linear Quadratic Regulator ....................... 24
      4.2.1 Controllability ................................ 24
   4.3 State estimation .................................... 25
      4.3.1 Kalman filter .................................. 25
   4.4 Linear Quadratic Gaussian ......................... 26
   4.5 Tracking with integral action .................... 28
   4.6 Control system layout .............................. 29
      4.6.1 Attitude controller ............................ 29
      4.6.2 Full model .................................... 30

5 Simulation and implementation ..................... 31
   5.1 Simulation Objectives .............................. 31
   5.2 Attitude controller ................................ 32
   5.3 Simulation of full model ......................... 33
      5.3.1 Simulation with LQR controller .............. 34
      5.3.2 Simulation of tracking and kalman filter ...... 36
   5.4 Evaluation .......................................... 37
5.5 Implementation ................................................. 37
  5.5.1 Generation of C code .................................... 37

6 Conclusion and further work .................................. 38

Bibliography ....................................................... 38

Figures ............................................................ 41

7 Appendix .......................................................... 42
1. Introduction

Ever since the first helicopter was invented it has never stopped to contribute to the everyday life around the world. As time has passed technological innovation have made it possible for every ordinary man to own a small helicopter and maneuver it by the help of a remote controller. Where the pilot in a full scale helicopter would make use of a onboard computer to control the movement a person with a RC helicopter would need to rely on his finger skills and vision to make sure the helicopter maintain the desired path via remote control from the ground. This has often shown to bee quite difficult for first time users and may in many cases lead to expensive crashes throughout a long period of training. It would save both money and time if it in some way would be possible to reduce the time spent on training by connecting something to the helicopter and thereby control and stabilize the helicopter for the user.

The idea is to investigate the possibility of constructing a stabilizer which reduces this complexity. A desired solution is to let the pilot control one movement in one direction or rotation around one axis while a microcontroller attached to the helicopters body controls the remaining degrees of freedom.

An important question is how this should be done and what kind of model this would be based on. It would be useful if it was possible to implement this control system on a onboard microcontroller in association with sensors and measuring devices.

This thesis is ment to give a novice reader a good overview over the complexity of controlling a helicopter. In chapter ?? the concept helicopter movement is explained in detail and the differential equations of motion is presented. In chapter ?? the differnt linearized models is explained and suitable contol methods is presented. A purposed simulation model is constructed. At last some simulaton results is presented in ?? along with a chapter about instrumentation and how one could implement this on a physical helicopter.
2. Concept of Helicopter movement

2.1 Introduction

To fully understand the concept of how to stabilize a helicopter it is important to obtain some knowledge related to helicopter movement. As any other moving object a helicopter will have a certain amount of degrees of freedom. A degree of freedom relates to a direction in a coordinate system which an object can move freely around or along. If an object is located in a three dimensional coordinate system $x,y,z$ movement along $x$ axis is one degree of freedom and angular movement around the same axis is another.

If a local coordinate system is placed in the middle of a RC helicopter (figure 2.1) it is common to have the $z$ axis pointing downwards, the $x$ axis to the right and $y$ to the left. The helicopter will have a total amount of six degrees of freedom. These are the translative movements along $x,y$ and $z$ axis as well as angular movement around each of them. The helicopter consists of different key parts which is connected to a remote controller which makes the movements possible.

![Figure 2.1: Helicopter with local coordinate system](image-url)
2.2 Movement related to the Z axis

The helicopter has two main parts which contributes to change of the movement. A main rotor is located at the top of the fuselage and is the main source to change the position along the z-axis. To lift off the ground or gain altitude the main rotor must generate more lift force than the force of gravity pulling it down.

![Main parts of Helicopter](image)

Figure 2.2: Main parts of Helicopter

At a distance from the main fuselage there is a tail rotor whichs main task is to control the angular movement around z.

![Angular movement around z](image)

Figure 2.3: Angular movement around z
The main and tail rotor consists of blades. These blades are shaped to be as aerodynamical as possible. The angle between the blade and the reference plane is called the angle of attack or pitch angle.

![Airflow](image)

Figure 2.4: Pitch angle

When the angle is increased or decreased for all blades on the main rotor along the whole rotation, the helicopter will change position along the z-axis more rapidly while usually maintaining the same thrust as its changes position. This feature is called collective pitch. The translative motion along z is denoted as $w$ and the angular $r$.

### 2.2.1 Cyclic Pitch

Where the collective pitch changes the pitch angle for all blades around the whole rotation the cyclic pitch changes the angle of the blades only for a certain part of the rotation. This leads to translative movement along the x and y axis.

![Cyclic pitch](image)

(a) Cyclic pitch in x direction  
(b) Cyclic pitch in y direction

Figure 2.5: Cyclic pitch

To make cyclic pitch possible the helicopters need a mechanism to transfer lateral and longitudinal stick input to change of cyclic pitch. This mechanism is located under the main rotor and is called a swashplate.
2.3 Movement related to X axis

When the helicopter fly forward the rotor disc tilted forward through the application of cyclic pitch - pitching the blades down on the advancing side and pitching up on the retreating side. This maneuver directs the thrust vector forward and applies a pitching moment around y axis (2.6) to the helicopter fuselage and accelerating the helicopter into forward flight. The translative motion along x is denoted as $u$ and the angular $p$.

![Figure 2.6: Angular movement around Y](image)

2.4 Movement related to Y axis

The same concept of angular and translative movement around X also happens when the helicopter is moving from side to side (2.7) The translative motion along y is denoted as $v$ and the angular $q$.

![Figure 2.7: Angular movement around Y](image)
2.4.1 Summary

$u$ and $\dot{v}$ describes the lateral and longitudinal motion of the fuselage.
$p$ and $\dot{q}$ describes the pitch and roll motion of the fuselage.
$w$ describes the vertical motion of the fuselage.
$r$ describes the fuselage yaw motion.

<table>
<thead>
<tr>
<th>Helicopter input</th>
<th>Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left/right cyclic</td>
<td>Roll</td>
</tr>
<tr>
<td>Forward/backwards cyclic</td>
<td>Forward/backward</td>
</tr>
<tr>
<td>Left/right tail</td>
<td>Yaw</td>
</tr>
<tr>
<td>Collective pitch/Throttle</td>
<td>Climb/dive</td>
</tr>
</tbody>
</table>
3. Modeling of the helicopter

3.1 Introduction

To fully understand the problem of how to model a helicopter in a simulation environment it is necessary to investigate all possible methods and the different approaches which can be made. In the book 'Helicopter Flight Dynamics' by Padfield [2] the construction of a helicopter dynamic model is explained in detail along with a linearized model. This modeling procedure is based on a full scale helicopter and it consists of quite complex terms. This model is used in 'Small-Size Unmanned Helicopter Guidance and Control' by Kurusu [13] and in 'Constructing and Simulating a Mathematical Model of Longitudinal Helicopter Flight Dynamics' by Fahad A Al Mahmood [14] to control the behavior of the helicopter. The problem with such a approach is the procedure which needs to be gone through to obtain all parameters needed. In [7] and [17] a detailed description of all forces and equations is given. [7] uses the nonlinear helicopter model presented by Padfield in addition to lead compensators and inner and outer control loops. The linearized model presented by Padfield is further adopted to small scaled helicopter by Mettler [3]. Mettlers approach is often used in project related to control and simulation of RC helicopters. Versions of this modeling method is shown to be valid in various reports and thesis. ( [16, 9, 8, 6] and [11]).

In this chapter the path from the nonlinear model by Padfield through to Mettlers version is explained to give a good description on how a suitable simulation model can be obtained. This chapter will present the helicopter model which later will be controlled. First the specifications of the helicopter will be defined. Second the nonlinear model, the linearized model and the simulation model used will be presented.

![Figure 3.1: Modeling process](image)

To simulate and try to control the behavior of the helicopter we need to construct a dynamical model which gives a better representation of the different aspects of helicopter movement.
3.2 Nonlinear model

3.2.1 Reference geometry

When constructing a nonlinear dynamical model of the helicopter it is important to denote the forces and moments in the correct way. Recalling that the translative velocities in relation to a local coordinate system is denoted as u,v,w and the angular movements p,q,r. This local coordinate system is also known as the body frame. In relation to the navigational frame the angular velocities is $\theta, \phi, \psi$ also known as pitch, roll and yaw.

![Figure 3.2: Reference axis](image)

To rotate a vector about a single axis it has to be multiplied by a transformation matrix. Rotation abot each of the three axis is given by:

$$
R_x(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \phi & \sin \phi \\
0 & -\sin \phi & \cos \phi
\end{bmatrix}
 \quad
R_y(\theta) = \begin{bmatrix}
\cos \theta & 0 & -\sin \theta \\
0 & 1 & 0 \\
\sin \theta & 0 & \cos \theta
\end{bmatrix}
 \quad
R_z(\psi) = \begin{bmatrix}
\cos \psi & \sin \psi & 0 \\
-\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

The total conversion from navigation frame to bodyframe is given by multiplying the three transformation matrices together:

$$
C^n_b = R_x(\phi)R_y(\theta)R_z(\psi)
$$

$$
C^n_b = \begin{bmatrix}
\cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\
\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & \sin \phi \cos \theta \\
\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta
\end{bmatrix}
$$
3.2.2 Translational movement

Newton second law states that:

\[ F = \frac{d}{dt}(mV_T) \] (3.2)

\( F \) = total external force applied to the body.
\( V_T \) = Total velocity of the body.
This can be further explained as:

\[ \frac{d}{dt}(V_T)_E = (\frac{d}{dt}V_T)_B + \omega \times V_T \] (3.3)

where \( \omega \times V_T \) is denoted as the Coriolis effect which is known as the motion resulting from the relative angular velocity of the moving frame with respect to the moving frame ([13])

In the body frame the vectors is expressed as

\[ \vec{V}_T = \vec{i}u + \vec{j}v + \vec{k}w \] (3.4)

\[ \vec{\omega} = \vec{i}p + \vec{j}q + \vec{k}r \] (3.5)

The total force \( \vec{F} \) acting on the helicopter fuselage is then

\[ \vec{F} = m \left\{ (\dot{u} + qw - vr)\vec{i} + (\dot{v} + ur - pw)\vec{j} + (\dot{w} + pv - uq)\vec{k} \right\} \] (3.6)

\( \vec{F} \) has components acting in each direction and can thereby be expressed as

\[ \vec{F} = \vec{i}F_x + \vec{j}F_y + \vec{k}F_z \] (3.7)

The gravity component will always point downwards in the navigation frame and this is therefore needed to be multiplied by the transformation matrix.

\[ F_{gb} = m \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \]

\[ F_{gb} = m \cdot g \cdot \begin{bmatrix} \sin(\theta) \\ -\sin(\phi)\cos(\theta) \\ \cos(\phi)\cos(\theta) \end{bmatrix} \]
Combining the two previous equations and adding the components of gravity gives us the following differential equations for the translational movement.

\[
\dot{u} = -wq + vr - g \cdot \sin(\theta) + \frac{F_x}{m} \tag{3.8}
\]
\[
\dot{v} = wp - ur + g \cdot \sin(\phi) \cos(\theta) + \frac{F_y}{m} \tag{3.9}
\]
\[
\dot{w} = uq - vp + g \cdot \cos(\phi) \cos(\theta) + \frac{F_z}{m} \tag{3.10}
\]

Where:

- \(u, v, w\) Translational velocity in x, y and z direction respectively
- \(p, q, r\) Angular velocity around x, y and z axis respectively
- \(\theta, \phi, \psi\) Angular velocity navigational frame (yaw, pitch, roll)
- \(m\) Mass of helicopter
- \(F_x, F_y, F_Z\) Forces acting in x, y and z direction respectively.

The forces \(\frac{F_x}{m}\) etc. will be presented as X, Y, Z since they consist of multiple forces acting on the helicopter.
3.2.3 Angular movement

Newtons second law on rotational form states:

$$\vec{M} = \frac{d}{dt}(\vec{H})$$  \hspace{1cm} (3.11)

$M=$total torque applied to the body. $H=$Total angular momentum.

This can be further explained as:

$$\vec{H} = I\vec{\omega}$$  \hspace{1cm} (3.12)

$$\frac{d}{dt}(\vec{H})_E = \frac{d}{dt}(\vec{H})_B + \vec{\omega} \times \vec{H}$$  \hspace{1cm} (3.13)

The total moment $\vec{M}$ in the body frame is expressed as:

$$\vec{M} = (\dot{p}I_{xx} + qr(I_{zz} - I_{yy}))\vec{i} + (\dot{q}I_{yy} + pr(I_{xx} - I_{zz}))\vec{j} + (\dot{r}I_{zz} + pq(I_{yy} - I_{xx}))\vec{k}$$  \hspace{1cm} (3.14)

The Moments $\vec{M}$ has components acting in each direction and can thereby be expressed as:

$$\vec{M} = \vec{i}M_x + \vec{j}M_y + \vec{k}M_z$$  \hspace{1cm} (3.15)

Rearranging the terms with respect to acceleration gives us:

$$\dot{p} = \frac{1}{I_{xx}}(-qr(I_{yy} - I_{zz}) + M)$$  \hspace{1cm} (3.16)

$$\dot{q} = \frac{1}{I_{yy}}(-pr(I_{zz} - I_{xx}) + N)$$  \hspace{1cm} (3.17)

$$\dot{r} = \frac{1}{I_{zz}}(-pq(I_{xx} - I_{yy}) + L)$$  \hspace{1cm} (3.18)

Where:

$p, q$ and $r$  \hspace{0.5cm} Angular velocity in x,y and z direction respectively

$I_{xx}, I_{yy}$ and $I_{zz}$  \hspace{0.5cm} Moments of inertia in x,y and z direction respectively

$L, M, N$ :  \hspace{0.5cm} Moments acting in x, y and z direction respectively.
Figure 3.3: Helicopter with moments and forces

The equations of motion for the helicopter related to the local coordinate system is now been explained in detail and can be summarized by figure?? . The moments in L, M and N can be further explained in the following way

\[
L = L_R + L_{TR} + L_f + L_{tp} + L_{fn} \\
M = M_R + M_{tp} + M_f \\
N = -Q_e + N_{vf} + N_{tf}
\]  

(3.19) \hspace{1cm} (3.20) \hspace{1cm} (3.21)

\[
X = X_{mr} + X_{fus} \\
Y = Y_{mr} + Y_{fus} + Y_{tR} + Y_{vf} \\
Z = Z_{mr} + Z_{fus} + Z_{ht}
\]  

(3.22) \hspace{1cm} (3.23) \hspace{1cm} (3.24)

This shows that the forces and moments have different components based on the given specifications of the helicopter.
The angular movement which now is expressed is the with respect to the body of the helicopter. We need to transform to earth related angular movement, better known as roll, pitch and yaw. These are represented as:

\[
\begin{align*}
\dot{\phi} &= p + \tan(\theta)(q \sin(\phi) + r \cos(\phi)) \\
\dot{\theta} &= q \cos(\phi) - r \sin(\phi) \\
\dot{\Psi} &= \sec(\theta)((q \sin(\phi)) + r \cos(\phi))
\end{align*}
\]  

Nine sets of differential equations have now been presented and concludes the explanation of the nonlinear model of the helicopter.

### 3.3 State space

As any other system the helicopter will give a output based on a given input. The helicopter is a multiple-input multiple-output (MIMO) system. A common way of represent a MIMO system is in the form of state space. This makes us able to represent our system in a compact form and makes the simulation procedure more visible. The shape of the system is given as:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du
\end{align*}
\]  

Where \(x\) is denoted as the internal variables of the system or states. \(Y\) equals the outputs of the system. Using the state space form makes it easier to modern control methods on bigger systems.

![State space block diagram](image-url)
3.4 Linearization

When we linearize a nonlinear differential equation, we linearize it for small signal inputs about the steady-state solution when the small-signal input is equal to zero.\[1\] The steady state equals the equilibrium of the system. It is common to say that we linearize the nonlinear equations with a linear equation for small excursions about the equilibrium point.

\[
f(x) - f(x_0) \approx \left. \frac{df}{dx} \right|_{(x=x_0)} (x - x_0) \tag{3.30}
\]

To make the linear system reflect the behavior of the nonlinear system we have to linearize around a trim point to be able to implement methods known from modern control theory. By a trim point we mean an operating point of the helicopter such as a specified torque input on the two rotors. It can be shown that linear control methods used with a linearized model only will function when applied with a constant reference value and not with a continuous function such as a sine or cosine function.

When a nonlinear system such as this is linearized it is assumed that the external X, Y, Z and moments L, M and N can be represented as analytic functions of the disturbed motion and their derivatives. As explained in section?? the forces and moments consist of different components. Using Taylor’s theorem for analytic functions gives as an example the terms for the force X.

\[
X = X_e + \frac{\delta X}{\delta u} \delta u + \frac{\delta X}{\delta v} \delta v + .. \tag{3.31}
\]

\[
\frac{\delta X}{\delta u} = X_u \tag{3.32}
\]
The states of the helicopter are defined as

\[ x = [u, v, p, q, \phi, \theta, a, b, w, r] \] (3.33)

To linearize our eight sets of equations we need to compute the steady state equations for each state. As an example we solve the lineariztion problem for the translational movement in x direction \( u \). As explained the equation for \( u \) is:

\[ \dot{u} = -wq + vr - g \cdot \sin(\theta) + \frac{F_x}{m} \] (3.34)

Rewriting terms which is depended on time gives:

\[ \frac{du}{dt} = -w(t) \cdot q(t) + v(t) \cdot r(t) + X - g \cdot \sin(\theta(t)) \] (3.35)

This equation can be further linearized using the method in...

\[ u = \frac{d}{du}|_{ss}(X) \cdot \bar{u} + \frac{d}{dv}|_{ss}(X + r_{ss}) \cdot \bar{v} + \frac{d}{dp}|_{ss}(X) \cdot \bar{p} + \frac{d}{dq}|_{ss}(X + r_{ss}) \cdot \bar{q} \] (3.36)

\[ -\frac{d}{d\phi}|_{ss}(g \cos \theta_{ss}) \cdot \bar{\phi} + \frac{d}{dw}|_{ss}(X + q_{ss}) \cdot \bar{w} + \frac{d}{dr}|_{ss}(X + v_{ss}) \cdot \bar{r} \] (3.37)

Doing this for all differential equations for all states gives an A matrix for the system. To see the full terms for each state see([2])

\[
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{p} \\
\dot{q} \\
\dot{\phi} \\
\dot{\theta} \\
\dot{w} \\
\dot{r}
\end{bmatrix} = 
\begin{bmatrix}
X_u & X_v & X_p & X_q & X_a & X_b & X_c & X_d & X_e \\
Y_u & Y_v & Y_p & Y_q & Y_a & Y_b & Y_c & Y_d & Y_e \\
L_u & L_v & L_p & L_q & L_a & L_b & L_c & L_d & L_e \\
M_u & M_v & M_p & M_q & M_a & M_b & M_c & M_d & M_e \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
Z_u & Z_v & Z_p & Z_q & Z_a & Z_b & Z_c & Z_d & Z_e \\
N_u & N_v & N_p & N_q & N_a & N_b & N_c & N_d & N_e
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
p \\
q \\
\phi \\
\theta \\
w \\
r
\end{bmatrix} = 
\begin{bmatrix}
X_w - Q_e & X_r + V_e & Y_w + P_e & Y_r - U_e \\
X_w - Q_e & X_r + V_e & Y_w + P_e & Y_r - U_e \\
L_w & L_v & L_p & L_q \\
M_w & M_v & M_p & M_q \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
Z_w & Z_r & Z_p & Z_q \\
N_w & N_r & N_p & N_q
\end{bmatrix}
\begin{bmatrix}
u \\
v \\
p \\
q \\
\phi \\
\theta \\
w \\
r
\end{bmatrix}
\]
3.4.1 Linearized for RC helicopter

The linearization method described in previous section can be shown to be simplified when we are working with a small scaled rotorcraft such as a RC helicopter. It is assumed that the linearization is done around an equilibrium point in hover where \([u, v, w] = [0, 0, 0]\).

Mettler purposes a method with more simplifications than Padfield. This because there are forces and moments which can be neglected and simplified when dealing with a small scaled rotorcraft in comparison to a full scale helicopter.

Notice how the terms for the translative and angular acceleration have been reduced.

\[
\begin{align*}
\dot{u} &= X_u u - g \phi + X_a \\
\dot{v} &= Y_v v + g \theta + Y_b b \\
\dot{p} &= 0 \\
\dot{q} &= 0 \\
\dot{\phi} &= p \\
\dot{\theta} &= q
\end{align*}
\]

\[
\begin{bmatrix}
\dot{u} \\
\dot{v} \\
\dot{p} \\
\dot{q} \\
\tau_f \dot{a} \\
\tau_f \dot{b} \\
\dot{w} \\
\dot{r}
\end{bmatrix} = 
\begin{bmatrix}
X_u & 0 & 0 & 0 & 0 & -g & X_a & 0 & 0 \\
0 & Y_v & 0 & 0 & -g & 0 & 0 & Y_b & 0 \\
L_u & -L_v & 0 & 0 & 0 & 0 & 0 & L_b & 0 \\
M_u & M_v & 0 & 0 & 0 & 0 & M_a & 0 & L_w \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} 
\begin{bmatrix}
u \\
v \\
p \\
q \\
a \\
b \\
w \\
r
\end{bmatrix} + 
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & Y_{ped} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
A_{lat} & A_{lon} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B_{lat} & B_{lon} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\delta_{lat} & \delta_{lon} & \delta_{ped} & \delta_{col}
\end{bmatrix}
\]

By inspection of the matrix we see a clear difference between this and the first more general method described in 3.4.
3.5 Simulation model

The method by Mettler however is designed for an advanced helicopter and it is necessary to simplify it further to make it suitable for testing. The yaw motion will be neglected since this doesn’t play a part in the complex dynamics and the yaw motion is often controlled by a gyro anyway.

Since the helicopter is in linearized in hover mode the helicopter is assumed to be without collective pitch. This to make the simulation process easier. Then the possible inputs and the complexity of the model have been reduced to only two inputs.
4. Helicopter Control

4.1 Introduction

To make the helicopter follow a desired path or maintain a certain heading or altitude it is necessary to implement some sort of control method. The control systems main task is to stabilize the plant and make sure it performs according to the given specification. It is also interesting to look at the complexity of the solution. If it is meant to be implemented on a microcontroller it would be useful if the solution doesn’t take up too much space or is too complex to execute.

To grasp the concept of this control problem it is important to get a good overview over the different approaches and the key differences between them. First we think of the helicopter as a full system with a given inputs and outputs. Recalling section?? that the helicopter has two inputs and a given number of outputs. One would think that the easiest way of controlling such a plant is by implementation of control methods such as PID control. The problem with such an approach is that PID is a single-input single-output (SISO) control method which makes it hard to implement caused by the fact that the helicopter consists of many different variables which are dependent on each other.

![Diagram](image)

Figure 4.1: Ways of controlling a plant
Extensive research has been done around the world to try to control the behavior with different approaches and results. In [5] Linear quadratic regulator (LQR) in combination with PID control is used along with an Extended Kalman filter. A LQR controller is also used in [14]. In [4], a controller is designed for attitude, heave, and yaw. Linear Quadratic Gaussian is used in [12] with setpoint tracking. This design is based on the prediction error method. Tracking of a reference point is implemented also in [16] and [11]. LQR and LQG along with tracking tend to be the most commonly used approaches, but there are people who use Lyapunov as in [15]. There are also approaches which result in bigger solutions such as [7] which controls a nonlinear helicopter model using lead compensators.
4.2 Linear Quadratic Regulator

The LQR controller is founded on three assumptions:
1) All states are available for feedback
2) All of the unstable modes are controllable
3) All unstable modes are observable

The LQR seeks to minimize the cost function:

$$ J = \int_0^\infty [x^T Q x + u^T R u] \, dt $$

(4.1)

(4.2)

Figure 4.2: LQR blocks

where Q and R are used to tune the performance of the controller.

4.2.1 Controllability

The LQR regulator will try to stabilize the plant or in fact place the poles of the system. If an input to a system can be found that takes every state variable from a desired initial state to a desired final state, the system is said to be controllable; otherwise the system is uncontrollable.[1]

The controllability matrix is given by:

$$ C_M = [B \ AB \ A^2B \ ... \ A^{(n-1)}B] $$

(4.3)

If the matrix $C_M$ is of rank $n$ the system is said to be completely controllable.[1] The rank equals the number of linearly independent rows or columns in the controllability matrix.
4.3 State estimation

4.3.1 Kalman filter

Kalman filter is used to estimate states in a given plant based on a mathematical model. The filter is a model-based algorithm used to estimate states which is under the influence of random noise both in the plant and related to the process measurements.

Figure 4.3: Block diagram of regulator with filter
4.4 Linear Quadratic Gaussian

\[
\dot{x} = Ax + Bu + wd
\]  
(4.4)

\[
y = Cx + Du + wn
\]  
(4.5)

The Linear Quadratic Gaussian (LQG) control method is a optimal controller for a linear system with white Gaussian noise. This type of controller is commonly used when the linear system is uncertain. Each state can be weighted by the quasratic matrix $Q$. The LQG method is a combination of a linear quadratic regulator(LQR) and a linear quadratic estimator(LQE). The objective of the the LQG is to minimize the cost function:

\[
J = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left[ x^TQx + u^TRu \right] dt \right\}
\]  
(4.6)

\[
Q = Q^T \geq 0
\]  
(4.7)

\[
R = R^T \geq 0
\]  
(4.8)

Figure 4.4: LQG concept

with the control $U(t) = -Kx(t)$ requires the availability of all states through process measurement. When the state variables are not accesible we can use $U(t) = -K\hat{x}$ where $\hat{x}(t)$ is an estimate of $x(t)$based on the output $y$.
The way of doing this is by repecatting the process dynamics. We construct a copy of the system on the form:

\[ \dot{\hat{X}} = A\hat{X} + Bu \]  \hspace{1cm} (4.9)

We define the state estimation error as \( \dot{e} = Ax - A\hat{x} = Ae \). If \( A \) is stable, the error will go to zero asymptotically. If \( A \) is unstable, \( e \) is unbounded and \( \hat{x} \) will grow further apart from \( x \).

To avoid this problem we consider a correction term where the output \( y \) is fed back to the estimator:

\[ \dot{\hat{X}} = A\hat{X} + Bu + L(y - \hat{y}) \]  \hspace{1cm} (4.10)
\[ \hat{x} = 0 \]  \hspace{1cm} (4.11)

where \( L \) is the observer gain matrix. The state estimation error is now

\[ \dot{e} = Ax - A\hat{X} - L(Cx - C\hat{x}) = (A - LC)e \]  \hspace{1cm} (4.12)
\[ e(0) = 0 \]  \hspace{1cm} (4.13)

The observer error will go to zero if \( L \) is chosen such that \( A-LC \) is stable.
4.5 Tracking with integral action

While the LQR stabilizes all states it does not guarantee that the given plant go to a desired reference signal. To do this it is necessary to introduce the use of integral action. This means a integrator and a gain is added in combination with the lqr controller. In addition to the states x it the sates z. The plant consiting of both the x and z is called the augmented plant. Matlab has a buildt in function $LQI$ which produces a big controller which consists of both the lqr and the gain integral action. The introduction of integral action to track a reference signals gives a state space model of

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
\dot{z} &= y - y_d
\end{align*}
\]  

\[ (4.14) \]  
\[ (4.15) \]

\[ \begin{bmatrix} \dot{X} \\ \dot{Z} \end{bmatrix} = X_{aug} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} X_{aug} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} u - \begin{bmatrix} 0 \\ 1 \end{bmatrix} y_d \]

This means that in order to get construct a controller for the new augmented plant we need to use $A_{new}$ and $B_{new}$ as system matrices. We wil then end up with a large controller and then we need to sepearte the new big controller into the lqr controller and the gain matrix used in the integral action.

Figure 4.5: Linear quadratic regulator with integral action
4.6 Control system layout

The RC helicopter has many states and variables. In order to develop a controller for a certain amount of degrees of freedom, the focus of this thesis has been to use a method which is easy to understand and is proven to work within the given specifications. A study of the literature mentioned results in a conclusion that the approach in [11] is a good way of controlling the helicopter. Some modifications have been done due to the parameters available and the desired results. The first thing which would be normal to assume is to adopt control algorithm directly on the model described in [16]. However, this is shown to be difficult caused by interaction of roll and pitch movement as shown in [16, 11].

4.6.1 Attitude controller

First a attitude controller is designed based on the model described in 3.4.1. Since it is decided that the system only will have input from the longitudinal and lateral input the terms related to these input is extracted into the B matrix.

\[
A_1 = \begin{bmatrix}
X_u & 0 & 0 & 0 & 0 & 0 \\
0 & Y_v & 0 & 0 & g & 0 \\
-L_u & -L_v & 0 & 0 & 0 & 0 \\
-M_u & M_v & 0 & 0 & 0 & 0 \\
0 & -g & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
B_1 = \begin{bmatrix}
A_{lon} & A_{lat} \\
B_{lo} & B_{lat}
\end{bmatrix}
\]

The \(C\) matrix is designed such that the controller have measurements from the roll and pitch angles, \(\theta\) and \(\phi\).

\[
C_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

The attitude controller is designed with LQR and tracking with integral action.

![Attitude model diagram](image)

Figure 4.6: Attitude model
4.6.2 Full model

The attitude controller is then implemented in the full model along with the lateral and longitudinal motion.

\[
A = \begin{bmatrix}
(A_1 - B_1 * K_r)^{12 \times 12} & 0^{8 \times 4} \\
G & 0^{2 \times 2} & G & 0^{2 \times 2} & V & 0^{2 \times 2} \\
0^{2 \times 2} & I & 0^{2 \times 4} \\
0^{2 \times 2} & 0^{2 \times 4}
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
0^{6 \times 2} \\
I \\
0^{4 \times 2}
\end{bmatrix}
\]

Where the \( G \) and \( V \) matrix is

\[
G = \begin{bmatrix}
-g & 0 \\
0 & g
\end{bmatrix},
\]

\[
V = \begin{bmatrix}
X_u & 0 \\
0 & Y_v
\end{bmatrix}
\]

The lateral and longitudinal motion then becomes:

\[
\dot{u} = X_u u - g(\theta + a) \tag{4.16}
\]

\[
\dot{v} = Y_v v + g(\phi + b) \tag{4.17}
\]

The \( C \) matrix makes sure the \( X \) and \( Y \) position of the helicopter is measured.

\[
C_1 = \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

Figure 4.7: Control of helicopter model
5. Simulation and implementation

5.1 Simulation Objectives

The objective in this thesis is to model and simulate a stabilizer for a RC helicopter. Recall that a model which tracks the lateral and longitudinal position of the helicopter is decided to be constructed with a control system consisting of a LQR regulator with integral action and disturbance filter. This control system have to be tested on a linear system of a helicopter before it can be implemented on a physical helicopter.

The model which is simulated is obtained in the described in section?? and the parameters obtained is located in [16].

The operating operating point for which the model was linearized have the following parameters:

Helicopter model: Raptor 90 SE.
Main rotor angular speed: 1250 Rpm
Tail rotor angular speed: 5000 Rpm.

The other parameters used can be found in the appendix.
5.2 Attitude controller

The augmented plant for the attitude controller is:

\[
a = \begin{bmatrix} A1 & 0 \\ -C1 & 0 \end{bmatrix}
\]

\[
b = \begin{bmatrix} B1 \\ 0 \end{bmatrix}
\]

This small part of the system is simulated to show how the pitch and roll movement can be implemented.

A controller for lqr and a gain for the tracking is constructed using the lqr command in Matlab on the augmented plant. The total controller \( K_r \) becomes:

\[
K_r = \begin{bmatrix}
16.1570 & 0.1264 & 3.2695 & -0.0041 & 15.2435 & -0.0140 & -31.6228 & 0.0299 \\
0.1518 & 35.3960 & 0.0027 & 3.0921 & 0.0145 & 14.8676 & -0.0299 & -31.6228
\end{bmatrix}
\]

The attitude controller is modelled in Matlab/simulink with the constructed controller. The integral action is implemented and tested for a given reference of 10 degrees for pitch angle and -10 degrees for roll which equals +-0.175 radians.

This results in a measurement of the tracked states which behaves satisfactory5.1.

![Figure 5.1: Results from attitude model](image-url)
5.3 Simulation of full model

The full scale model is simulated with initial values for speed in x and y direction of 0.001. This is done to not make the simulation crash and to create a more life like environment. First it useful to check the response of the system without any control and with just measurements of the positions.

Figure 5.2: Position with no control

Figure 5.2 shows an uncontrolled response which behaves as expected. The helicopter just moves far away in both directions.
5.3.1 Simulation with LQR controller

A LQR controller is constructed along with a gain for integral action using the `lqi` command in Matlab. First a test with only LQR controller and a random reference is tested. The LQR gain $F_1$ is found to be

$$F_1 = \begin{bmatrix} 0.4039 & 0.0019 & 0.0323 & -0.0001 & 1.1473 & -0.0002 & 10.2313 & 0.0007 & 1.6191 & 0 \\ 1.1532 & 0 & 0.0009 & 0.4267 & -0 & 0.0248 & -0.0004 & 0.9909 & 0.0007 & 10.5470 & 0 & -1.6 \\ 0 & -1.11597 & \end{bmatrix}$$

The response in the position output(5.3) along with the control signal(5.4) has satisfactory results. The output becomes stable at a given reference.

![Figure 5.3: Position](image)

![Figure 5.4: Control signal lqr](image)
Stability

An important concept of LQR control is to make the system become stable. A bode plot is constructed to show the frequency response with respect to stability. This also shows the gain and phase margins of the system.

Figure 5.5: Bode plot of closed vs open loop

35
5.3.2 Simulation of tracking and kalman filter

The LQR gain is combined with a kalman filter and tracking with integral action. (5.6) A reference signal which goes to 10 meters after 20 seconds is given to the X positon reference while a reference of 2 meters is given to the y positon. It is clear that the respons of the system is quite slow. This has to do with the fact that the angle of the swashplate (longitudinal and lateral angel) must not exceed +10 degrees [11]. To make this happend it was necesarry to adjust the Q matrix accoringly and have penalize the states for X and Y position with a value of 0.01. The R matrix has the form of a two by to identity matrix with 0.01 as a multiplied gain.
5.4 Evaluation

The control system which is tested is based on a model presented in [11] and the parameters used is from [16]. This may create some deviation from the results which has been achieved in [11]. The controller can not exceed ±-10 degrees and this makes the system follow a reference slowly.

5.5 Implementation

The regulation and simulation of the linearized model may seem easy to control on the computer. Another thing is to actually implement it and test it on a physical helicopter. Observer design have to be taken into consideration. The solution presented in this thesis assumes that all states are measurable.

If sensors were to be implemented there may occur problems in relation to electromagnetic interference from motor and speed controller if such is implemented [5]. This interference may cause the microcontroller to periodically reset.

5.5.1 Generation of C code

To make use of the regulation algorithm it is necessary to generate C code from Matlab. This can be done in different ways and depends on the way the code is made in Matlab. Both methods is additional programming tools available with Matlab. One way is to use Matlab Coder which is makes it possible to generate C code directly from Matlab algorithms. This however demands that your Matlab program only consists of code in a script. Another way which may be more useful is use of the Simulink Coder. This add-on generates C code from Simulink blocksets and makes it ready for a real-time target.
6. Conclusion and further work

The purpose of this thesis has been to investigate the possibilities of constructing a stabilizer for a RC helicopter by implementation of various control methods. A literature study has been performed in order to understand the control problem of helicopter behaviour. A solution has been developed for parameters given for a Raptor 90 in hover. A solution for tracking of position reference in x and y direction have been implemented and proven to have satisfactory results. How to implement such a solution on a physical helicopter have been discussed.

This thesis presents good opportunities for further work and implementation possibilities for future projects. One approach is to start with the linearized model presented by Mettler 3.4.1 in [3]. The parameters in this model can be determined through experiments. This can however be quite difficult and require a high degree of technical insight. [3] and [16] presents parameters for a range of RC helicopter types. Control of these models can be tested in Matlab/simulink and implemented on a microcontroller with methods described in this thesis. It might be useful to construct a testbench such as the one described in [10] which reduces the degrees of freedom and makes sure that the helicopter stays within a controlled environment.


# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Helicopter with local coordinate system</td>
<td>6</td>
</tr>
<tr>
<td>2.2</td>
<td>Main parts of Helicopter</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Angular movement around z</td>
<td>7</td>
</tr>
<tr>
<td>2.4</td>
<td>Pitch angle</td>
<td>8</td>
</tr>
<tr>
<td>2.5</td>
<td>Cyclic pitch</td>
<td>8</td>
</tr>
<tr>
<td>2.6</td>
<td>Angular movement around Y</td>
<td>9</td>
</tr>
<tr>
<td>2.7</td>
<td>Angular movement around Y</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Modeling process</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Reference axis</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>Helicopter with moments and forces</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>State space block diagram</td>
<td>17</td>
</tr>
<tr>
<td>4.1</td>
<td>Ways of controlling a plant</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>LQR blocks</td>
<td>24</td>
</tr>
<tr>
<td>4.3</td>
<td>Block diagram of regulator with filter</td>
<td>25</td>
</tr>
<tr>
<td>4.4</td>
<td>LQG concept</td>
<td>26</td>
</tr>
<tr>
<td>4.5</td>
<td>Linear quadratic regulator with integral action</td>
<td>28</td>
</tr>
<tr>
<td>4.6</td>
<td>Attitude model</td>
<td>29</td>
</tr>
<tr>
<td>4.7</td>
<td>Control of helicopter model</td>
<td>30</td>
</tr>
<tr>
<td>5.1</td>
<td>Results from attitude model</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>Position with no control</td>
<td>33</td>
</tr>
<tr>
<td>5.3</td>
<td>Position</td>
<td>34</td>
</tr>
<tr>
<td>5.4</td>
<td>Control signal LQR</td>
<td>34</td>
</tr>
<tr>
<td>5.5</td>
<td>Bode plot of closed vs open loop</td>
<td>35</td>
</tr>
<tr>
<td>5.6</td>
<td>Response of full model</td>
<td>36</td>
</tr>
<tr>
<td>5.7</td>
<td>Control signal</td>
<td>36</td>
</tr>
<tr>
<td>5.8</td>
<td>Full scale tracking model with kalman filter</td>
<td>37</td>
</tr>
</tbody>
</table>
7. Appendix
clc  
clear all  
close all  

N=2.982;  
Ab=0.773;  
Ba=0.618;  
Lb=1172.4817;  
Lu=-0.0244;  
Lv=-0.1173;  
Ma=307.571;  
Mu=0.2542;  
Mv=-0.06013;  
Np=0;  
Nr=-10.71;  
N=2.982;  
Nw=-0.7076;  
taufi=30.71;  

Xa=9.389;  
Xu=-0.03996;  
Yb=-9.389;  
Yv=-0.05989;  
Za=0;  
Zb=0;  
Zr=0;  
Zw=-2.055;  
g=-9.389;  

Alo=4.059;  
Alat=-0.01610;  
Blo=-0.01017;  
Blat=4.085;  
Zcol=-13.11;  
Ncol=3.749;  
Nped=26.90;  

A1=[(-taufi) Ab -1 0 0 0;...  
    Ba (-taufi) 0 -1 0 0;...  
    Ma 0 0 0 0 0;...  
    0 Lb 0 0 0 0;...  
    0 0 1 0 0 0;...  
    0 0 0 1 0 0];  
B1=[Alo Alat;Blo Blat;0 0;0 0;  
    0 0;0 0];  

C1=[0 0 0 1 0;...  
    0 0 0 0 1];  

D1=zeros(2,2);  
systest=ss(A1,B1,C1,D1)
\[
\text{Creating Augmented plant--}
\]
\[
a = [A_1 \, \text{zeros}(6, 2); -C_1 \, \text{zeros}(2, 2)];
\]
\[
b = [B_1; \text{zeros}(2, 2)];
\]
\[
c = \text{eye}(\text{size}(a));
\]
\[
d = \text{zeros}(\text{size}(b));
\]
\[
sys = \text{ss}(a, b, c, d);
\]
\[
[n, m] = \text{size}(B_1);
\]
\[
Q_{\text{new}} = \text{eye}(\text{size}(a));
\]
\[
Q_{\text{new}}(7, 7) = 100;
\]
\[
Q_{\text{new}}(8, 8) = 100;
\]
\[
R_{\text{new}} = \text{eye}(m, m) \times 0.1;
\]
\[
K_r = \text{lqr}(a, b, Q_{\text{new}}, R_{\text{new}});
\]
\[
K_{rp} = K_r(1:m, 1:n);
\]
\[
K_{ri} = K_r(1:m, n+1:n+2);
\]
\[
\text{Creating big augmented plant--}
\]
\[
G = [-g \, 0; 0 \, g];
\]
\[
I = \text{eye}(2, 2);
\]
\[
V = [X_u \, 0; 0 \, Y_v];
\]
\[
A = [a - b \times K_r \, \text{zeros}(8, 2) \; \text{zeros}(2, 2) \; \text{zeros}(2, 2) \; \text{zeros}(2, 2); \; \text{zeros}(2, 8) \; I \; \text{zeros}(2, 2)];
\]
\[
B = [\text{zeros}(2, 6) \; \text{eye}(2, 2) \; \text{zeros}(2, 4)];
\]
\[
C = [0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 1 \; 0; \; 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 1];
\]
\[
D_{\text{top}} = \text{zeros}(2, 2);
\]
\[
Q_{\text{new top}} = \text{eye}(14, 14);
\]
\[
Q_{\text{new top}}(14, 14) = 0.001;
\]
\[
Q_{\text{new top}}(13, 13) = 0.01;
\]
\[
Q_{\text{new top}}(10, 10) = 1;
\]
\[
Q_{\text{new top}}(9, 9) = 1;
\]
\[
R_{\text{new top}} = \text{eye}(2, 2) \times 0.01;
\]
\[
\text{SYStop} = \text{ss}(A, B, C, D_{\text{top}});
\]
\[
[K_{\text{test}}, S_{\text{test}}, \text{etest}] = \text{lqi}(\text{SYStop}, Q_{\text{new top}}, R_{\text{new top}});
\]
\[
F_2 = K_{\text{test}}(1:2, 13:14);
\]
\[
F_1 = K_{\text{test}}(1:2, 1:12);
\]
\[
[K_{\text{full}}, L_{\text{full}}, F_{\text{ful}}] = \text{kalman}(\text{SYStop}, \text{eye}(2, 2) \times 0.001, \text{eye}(2, 2) \times 0.001);
\]
\[
[\text{NUM}, \text{DEN}] = \text{ss2tf}(A, B, C, D_{\text{top}}, 2);
\]
\[
[\text{NUM1}, \text{DEN1}] = \text{ss2tf}((A - B \times F_1), B, C, D_{\text{top}}, 2);
\]
\[
\text{Openloop} = \text{tf}(\text{NUM}(1,:), \text{DEN});
\]
\[
\text{Closedloop} = \text{tf}(\text{NUM1}(1,:), \text{DEN1});
\]
%-------------------Plots-------------------
figure
sim('helicopter_main_lqr')
plot(t.signals.values, Pos.signals.values(:,1), ('r--')); hold on
plot(t.signals.values, Pos.signals.values(:,2), 'b--'); hold on
sim('helicopter_main_lqr')
plot(t.signals.values, Pos.signals.values(:,1), ('r')); hold on
plot(t.signals.values, Pos.signals.values(:,2), ('b'));
legend('X pos no LQR', 'Y pos no LQR', 'X pos with LQR', 'Y pos with LQR')
axis([0, 3, -0.08, 0.1])
xlabel('time[sec]')
ylabel('Position [m]')
grid on
hold off

figure
sim('helicopter_main_lqr')
plot(t.signals.values, Pos.signals.values(:,1), ('r--')); hold on
plot(t.signals.values, Pos.signals.values(:,2), 'b--');
xlabel('time[sec]')
ylabel('Position [m]')
grid on

figure
grid on
bode(Openloop)
hold on
bode(Closedloop)

figure
sim('helicopter_main_lqr')
plot(t.signals.values, Control.signals.values(:,,:), ('g')); hold on
plot(t.signals.values, Control1.signals.values(:,,:), ('b'));
legend('Control signal no LQR', 'Control signal no LQR', 'Control signal LQR', 'Control signal LQR')
axis([0, 0.3, -0.08, 2])
xlabel('time[sec]')
ylabel('Position [rad]')
grid on

figure
sim('WORKING_helitrack')
plot(t.signals.values, Pos.signals.values(:,1), ('r'));
hold on
plot(t.signals.values,Ref.signals.values(:,1),'r--');
hold on
plot(t.signals.values,Pos.signals.values(:,2),'b');
hold on
plot(t.signals.values,Ref.signals.values(:,2),'b--');
legend('X pos with tracking','X pos reference','Y pos with tracking','Y pos reference')
xlabel('time[sec]')
ylabel('Position [m]')
grid on

figure
sim('WORKING_helitrack')
plot(t.signals.values,Control.signals.values(:,1),'k');
hold on
plot(t.signals.values,Control.signals.values(:,2),'c');
legend('Control signal phi','Control signal theta')
xlabel('time[sec]')
ylabel('Reference [rad]')
grid on