# 3-DOF Helicopter Laboratory Session

O. Wigström*

Department of Signals and Systems
Chalmers University of Technology

25th November 2014

## 1 Introduction

This document is geared towards students taking the 'Linear control systems design' (SSY285) course who are to participate in a laboratory session involving a 3-DOF helicopter. Contained within this document is background information related to the laboratory equipment, preparation exercises and also a number of exercises which should be carried out before (preparation) and during the session. Students are expected to read the entire document as well as successfully perform all the preparation exercises in Section 2 before attending the session. Failure to prepare for the lab session in advance may result in rejection from the laboratory session.

The scope of this document is to prepare and guide for a laboratory session involving a 3-DOF helicopter. Note that all exercises related to the laboratory session are carried out in continuous time.

The purpose of the preparation exercises is to train you how computer software can be used for model analysis and control design. It will also familiarize you with the process and prepare you for the experimental tasks. Before the laboratory session you should linearize a given nonlinear system model. You will also test the validity of your linearized model compared to the nonlinear as well as perform control design.

For the actual laboratory experiments, you will use LQ and LQI principles to control the process level axis, travel axis and travel axis speed, as well as combinations thereof. The purpose of the laboratory experiments are to provide you with some practical experience in applying the methods studied in this course on a real process.

*oskar.wigstrom@chalmers.se

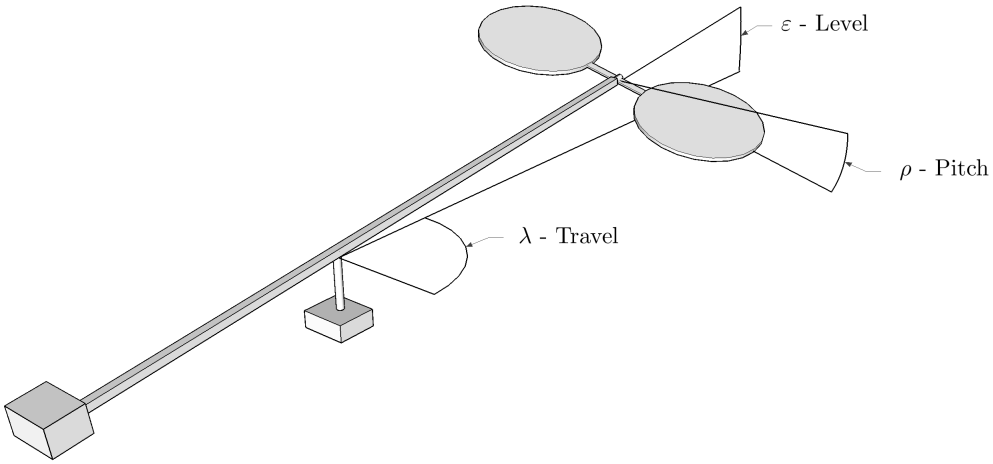


Figure 1: 3-DOF helicopter layout.

### 1.1 Model

Figure 1 depicts the layout of the helicopter device. The three angular position states we will use for our model are the level ($\varepsilon$), pitch ($\rho$) and travel ($\lambda$) angles. The input voltage for the front and back motor will be denoted by $V_f$ and $V_b$, respectively. A positive voltage implies air being blown downwards creating an upwards lift. A simplified nonlinear model of the process is given by the following input-affine system

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where

$$x = [\varepsilon, \rho, \lambda, \dot{\varepsilon}, \dot{\rho}, \dot{\lambda}]^T, \tag{2}$$

$$u = [V_f, V_b]^T. \tag{3}$$

The function $f(x)$ is defined as

$$f(x) = \begin{bmatrix} \dot{\varepsilon} \\ \dot{\rho} \\ \dot{\lambda} \\ p_1 cos\varepsilon + p_2 sin\varepsilon + p_3\dot{\varepsilon} \\ p_5 cos\rho + p_6 sin\rho + p_7\dot{\rho} \\ p_9\dot{\lambda} \end{bmatrix}, \tag{4}$$

and g(x) as

$$g(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ p_4 cos\rho & p_4 cos\rho \\ p_8 & -p_8 \\ p_{10} sin\rho & p_{10} sin\rho \end{bmatrix}. \tag{5}$$

The weights $p$ are defined as

$$\begin{aligned} p_1 &= [-(M_f + M_b)gL_a + M_c gL_c]/J_\varepsilon & p_2 &= [-(M_f + M_b)g(L_d + L_e) + M_c gL_d]/J_\varepsilon \\ p_3 &= -\eta_\varepsilon/J_\varepsilon & p_4 &= K_m L_a/J_\varepsilon \\ p_5 &= (-M_f + M_b)gL_h/J_\rho & p_6 &= -(M_f + M_b)gL_e/J_\rho \\ p_7 &= -\eta_\rho/J_\rho & p_8 &= K_m L_h/J_\rho \\ p_9 &= -\eta_\lambda/J_\lambda & p_{10} &= -K_m L_a/J_\lambda \end{aligned} \tag{6}$$

In Table 1.1, you can find the physical quantities used for the weightings in (6).

| Symbol | Value | Description |
|---|---|---|
| $M_f$ | 0.71 kg | Mass of front propeller assembly |
| $M_b$ | 0.71 kg | Mass of back propeller assembly |
| $M_c$ | 1.69 kg | Mass of the counterweight |
| $L_d$ | 0.05 m | The length of pendulum for elevation axis |
| $L_c$ | 0.44 m | Distance from pivot point to counterweight |
| $L_a$ | 0.62 m | Distance from pivot point to helicopter body |
| $L_e$ | 0.02 m | The length of pendulum for pitch axis |
| $L_h$ | 0.18 m | The distance from the pitch axis to motor |
| $g$ | 9.81 m/s$^2$ | Gravitational acceleration |
| $K_m$ | 0.12 N/V | Propeller force-thrust constant |
| $J_\varepsilon$ | 0.86 kg m$^2$ | Moment of inertia about the level axis |
| $J_\rho$ | 0.044 kg m$^2$ | Moment of inertia about the pitch axis |
| $J_\lambda$ | 0.82 kg m$^2$ | Moment of inertia about the travel axis |
| $\eta_\varepsilon$ | 0.001 kg m$^2$/s | Coefficient of viscous friction, level axis |
| $\eta_\rho$ | 0.001 kg m$^2$/s | Coefficient of viscous friction, pitch axis |
| $\eta_\lambda$ | 0.005 kg m$^2$/s | Coefficient of viscous friction, travel axis |

## 1.2 Control design in MATLAB

This section will provide a quick review of the linear control methods you have been taught in the course. A continuous linear time system model is

given by

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{7}$$

where a control law under the form of

$$u = -Kx + K_r r \tag{8}$$

is assumed which minimizes the cost criteria

$$J = \int_0^\infty (x^T Q_x x + u^T Q_u u) dt \tag{9}$$

where, the gain matrix $K$ can be computed in Matlab by

```
K = LQR(A,B,Q_x,Q_u)
```

The steady state output for a step of size $r_0$ can be computed by the final value theorem as,

$$\lim_{t\to\infty} y(t) = \lim_{s\to 0} C(sI - A + BK)^{-1} BK_r r_0 \tag{10}$$

If the system reaches $r_0$ at steady state, then

$$\lim_{s\to 0} C(sI - A + BK)^{-1} BK_r = I. \tag{11}$$

Given a square $K_r$, the previous equation can be solved for $K_r$.

Another approach for asymptotic reference tracking is to integrate the tracking error $(y-r)$. For integral action, the system matrices (7) are augmented with integral states. If for example we would like an integral action at the plant outputs, then the augmented system is given by

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{x}_I \end{bmatrix} &= \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} K_r \\ -I \end{bmatrix} r \\ &= A_I x_a + B_I u + \hat{K}_r r. \end{aligned} \tag{12}$$

The optimal gain $K_I$ in

$$u = -[K_1 \; K_2] \begin{bmatrix} x \\ x_I \end{bmatrix} = -K_I x_a \tag{13}$$

which minimizes

$$J = \int_0^\infty (x_a^T Q_{Ix} x_a + u^T Q_{Iu} u) dt \tag{14}$$

is now given by

```
K_I = LQR(A_I, B_I, Q_Ix, Q_Iu).
```

## 1.3 Simulating nonlinear systems in Matlab

Trying out the linear controller with linear model is simple in Matlab,

```
sys = ss(A-B*L,B,C,D);
step(sys);
```

If we would like to try our linear controller on the nonlinear model however, we have to do it a bit differently. Either you can model the system using Simulink (recommended), or you can use an ordinary differential equation (ODE) solver as suggested here in the following Matlab code:

```
function [] = mySimulation()

% Declare any constants etc which are to be passed to the model
data = ...

% End time and initial conditions
T = ...
X0 = ...

% Run ODE solver, ODE45 for example
[t,y] = ode45(@(t,y) myFunc(t,y,data), [0, T], X0)

%Plot results
plot(t,y)

end


% This function should return the derivative of x
function dx = myFunc(t, y, data)

% In the linear case for example
u = data.L*x;
dx = data.A*x + data.B*u;

end
```

## 1.4 Running the laboratory equipment via Simulink

These instructions will guide you through the process of running experiments in the laboratory. First, log on to the lab computer using the login and

password supplied on site. Open up the laboratory project folder located on the desktop, containing sveral `.m` and `.sim` files.

The procedure for running the experiments is as follows. Open up the `.m` and `.sim` files specified later in this laboratory guide. Modify the `.m` file with your own data and run the file. Next, with the `.sim` file open, choose 'Connect to target' within the 'Simulation' menu in Simulink. Before your first run, ask the laboratory assistant for approval. Later, if you at any point feel unsure, ask the assistant for help. Finally, select 'Run' from the same menu to start the process. To stop, simply press the stop button.

It is **very important** that you at **all times** are ready to physically intervene in case the process and the controller turns out to be unstable. There is also an **emergency stop** button which will cut power to the motors when pressed, please use this option or the stop button at any sign of trouble. Also, if turned of at a high altitude, make sure to soften the fall by reaching out and grabbing the helicopter with your hand.

# 2 Preparation

As this course focuses on linear control methods, you will have to linearize the system in order to perform control design. It is also important that you try out your controller in simulation a priori the laboratory session such as to test the performance and validity of your controller.

Your first task is to linearize the model given by equations (7)-(6) and Table 1.1

**Task 2.a (Linearization)** A Simulink model describing the nonlinear system is available on the course home page. Use the Linear Analysis Tool in Simulink to derive a linearized model. The point around which the system is linearized can be set to "Model Initial Condition". We would like to linearize the system at the origin, that is, set the initial condition: $x_0 = [0\ 0\ 0\ 0\ 0\ 0]^T$. As for the inputs $u_0 = [V_{f0}, V_{b0}]$, at the origin it is reasonable to assume $V_{f0} = V_{b0}$. To find $V_{f0}$, simply solve $f(x_0) + g(x_0)u_0 = 0$ for $u_0$.

**Task 2.b (Open loop dynamics)** You now have both a linear and a nonlinear model. It is important to investigate how well the linear model represents the nonlinear one. Simulate both models using an all zero initial state and unit input on input one. Analyze the output, what happens to the various states? Compare the results, how do the linear and nonlinear models' reactions differ?

.....

.....

.....

**Task 2.c (Closed loop dynamics part.1)** Generate a continuous time LQ controller based on the system linearized at the origin. Make sure to add $u_0$ to the control input of the nonlinear model such it rests at steady state. Try out your controller on both the linear and nonlinear system. Use reference steps on $\varepsilon$ and $\lambda$ separately and pay specific attention to their outputs. Comment on your results. Remember that your state space model uses radians!

.....

.....

.....

# 3 Laboratory session

## 3.1 Level control

In this part of the laboratory session you will examine how regulation of the helicopter's level works based on your linearized model.

For all exercises, the program requires you to specify four matrices: $K \in \mathbb{R}^{2\times n}$, where $n$ is the number of states, $K_I \in \mathbb{R}^{2\times 2}$, $K_r \in \mathbb{R}^{2\times 2}$ and $V_K \in \mathbb{R}^2_+$, which represent the proportional gain matrix, integral gain matrix and voltage bias respectively. The latter is simply a constant voltage added to the motors.

The first step is to use pure LQ-control. At the bottom om `lab_init.m` you will find the following code

```
% Fetch linearized system
X0 = [];
[A,B] = linnlsys(X0);

% Define weighting matrices and compute L
Q_x = [];
Q_u = [];

K = LQR(A,B,Q_x,Q_u);
```

```
K_I = [];
V_K = [];

C = [];
K_r = [];
```

besides setting up the regulator, `lab_init.m` will also set other necessary settings. In these first experiments, you will use `lab_elev_travel.mdl` to control the process. Each time you modify any constants, make sure you run the modified file and perform the necessary steps in Simulink explained in Section 1.4.

**Task 3.1.a (Setting up a simple LQ controller)** Now, fill in the blank matrices and run the code. You should set $K_I$ and $V_K$ to zero for this first exercise. In Simulink, make sure the level set point is 0. Use only as much weight on $\rho$ and $\lambda$ as is needed to keep the travel movement to a minimum and set $Q_u$ to identity. In this section, the system has two outputs, $\varepsilon$ and $\lambda$, define $C$ such as to model this. Also compute $K_r$ using equation (11).

**Task 3.1.b (LQ control of level)** Start with a conservative weight for $\varepsilon$, progressively choose a higher weight and record your results. For each attempt record the weight, steady state level and steady state control signal.

.....

.....

.....

As you may have noticed in the previous exercise, the steady state error is never completely eliminated. We will try to add a base voltage $V_K$ to the control signal. This means that without any regulator control action, the propellers should deliver enough thrust for the helicopter to reach zero level. The control systems main task will be to steer the helicopter in the immediate area close to zero level.

**Task 3.1.c (Estimating the required base voltage)** Using the data from the previous exercise, make a linear or second order approximation of of the control signal to level gain. Extrapolate your results and determine the control signal needed to reach zero level. Run the process and record its behaviour.

.....

. . . . . .

. . . . . .

**Task 3.1.d (LQ control of level with added base voltage)** Now try exchanging the constant level reference for a square wave which alternates between $-10°$ and $10°$. Is the steady state error eliminated, and why?

. . . . . .

. . . . . .

. . . . . .

As you should have noticed, the helicopter steers very close to zero level, but it does not work as well for the square wave reference. We will keep the base voltage and use integration to remove the small steady state error.

In the bottom of `lab_init.m`, just after the linearized system has been loaded, extend the $A$ and $B$ matrices to include an integral state for the level tracking error. Do not forget to also extend $Q_x$ as well. The resulting $\tilde{K}$ matrix can be split into proportional and integrative parts as follows:

```
K = LQR(A,B,Q_x,Q_u);
K_I = [K(:,7) zeros(2,1)];
K = K(:,1:6);
```

**Task 3.1.e (Selecting an appropriate feed forward term)** What is a reasonable choice for $K_r$ now that we are using LQI? Discuss among yourselves and present your hypothesis to the lab assistant.

. . . . . .

. . . . . .

. . . . . .

**Task 3.1.f (LQI control of level, with added base voltage)** Start out with the same weights for the states as in the previous exercise, run the system and iterate the values of the integral state weights. Use the same square wave as before. What values for $Q_{Ix}$ do give good results?

. . . . . .

. . . . . .

. . . . . .

Even though you get good results in the previous exercise, those results are only good around in the area close to zero level.

**Task 3.1.g (Robustness for larger set point changes)** With your preferred settings from the previous exercise, try a square wave which alternates between $-20°$ and $20°$ instead. How do your results look now?

. . . . . .

. . . . . .

. . . . . .

We will now remove the base voltage and try to control the system using only the regulator. Set $V_K$ to zero.

**Task 3.1.h (LQI control of level)** Go back to the smaller square wave and try different settings for the weighting matrices. Once you've found a good settings, try this for the larger square wave as well and compare your result to the ones from Task 3.1.f and 3.1.g.

. . . . . .

. . . . . .

. . . . . .

**Task 3.1.i (Robustness)** You will now try adding a disturbance to the system. Use one of the weights supplied by the laboratory assistant. Start by using the LQ controller with added base voltage from Task 3.1.d. Apply the weight to the system and see how it reacts to the disturbance. Next, use the controller from Task 3.1.h and compare the results.

. . . . . .

. . . . . .

. . . . . .

## 3.2 Travel angle control

In this part of the lab, you will attempt to control the position of the helicopter around the travel axis. For travel control, there is really no use for

a base voltage as the steady state input is independent of the travel angle. We will consider LQ and LQI in this section.

Use the weighting matrices and code from Task 3.1.h. This means that you will have integral action for the level but only proportional control for the travel angle $\lambda$.

**Task 3.2.a (LQ control of travel angle)** Apply a constant level reference and a square wave reference of amplitude $\pm 10^\circ$ for $\lambda$. Try out different settings for the weights related to $\lambda$ and $\rho$, let the weights related to level stay as they are. Can you find any good settings? Is there a steady state error?

.....

.....

.....

Extend the $A$ and $B$ matrices such that there are integral states for both $\varepsilon$ and $\lambda$. Remember to do the necessary changes to $Q_x$ as well. You should now split the resulting $K$ matrix as:

```
K = LQR(A,B,Q_x,Q_u);
K_I = K(:,7:8);
K = K(:,1:6);
```

**Task 3.2.b (LQI control of travel angle)** Start with a conservative value and try to find a good weighting matrix. How does the resulting system behave?

.....

.....

.....

**Task 3.2.c (Robustness)** You should now evaluate the robustness of your angle and travel control. Instead of using the weight, give the system a gentle push with your hand. How does the system react?

.....

.....

.....

### 3.3 Travel speed control (if time permits)

You will now attempt to move the helicopter at a constant speed around the travel axis. Close down your current Simulink file and open up `lab_speed.mdl`. We will now consider $\varepsilon$ and $\dot{\lambda}$ as system outputs. Given a constant set point for $\dot{\lambda}$, $\lambda$ will in turn grow unboundedly. Thus we will have to remove any trace of $\lambda$ (not $\dot{\lambda}$!) from our model. Modify $A$, $B$, $C$ and $Q_x$ accordingly.

**Task 3.3.a (LQ control of travel speed )** Start out with a base voltage and settings as the ones you used in Task 3.1.h, with the modifications above added of course. Use a constant elevation of $0^\circ$ and set a constant speed reference of $10^\circ/s$. Try out different weights for $\dot{\lambda}$ using varied speed set points and record the system performance.

.....

.....

.....

**Task 3.3.b (LQI control of travel speed)** Add integral action to $\dot{\lambda}$, how does the system perform now? Change the reference speed.

.....

.....

.....

**Task 3.3.c (Varying travel reference speed)** Using your preferred settings from the previous exercise, apply a square wave to the speed reference such that the $\dot{\lambda}$ reference switches between say $15^\circ/s$ and $30^\circ/s$.

.....

.....

.....

**Task 3.3.d (Varying both reference travel speed and level)** Using the same reference for $\dot{\lambda}$ as in the previous exercise, add a square wave to the

$\varepsilon$ reference. Let the square wave have amplitude 10°. Observe the system behaviour, are you satisfied with its performance?

.....

.....

.....