Protocol for "Praktikum Konzept der Regelungstechnik"

Yanpeng Mei, **Azer Messaoudi**, **Zhiming Ma**

Abstract: This report introduces our solution to the task in the "Praktikum Konzepte der Regelungstechnik" at the IST. To design a suitable controller of the QUANSER 3-DOF Helicopter, a simplified nonlinear model is implemented and the linearization of this model is performed. Together with a reasonable desired trajectory, a LQI controller with pre-filter is designed to make sure the helicopter can meet the overall mission requirements and fulfill the constraints.

1. INTRODUCTION

1.1 Mission Description

The underlying control problem is the cargo-transportation with the QUANSER 3-DOF helicopter as shown in Fig. 1, of which the "Black Box Model" is given.

Our task is to make sure that the helicopter can finish the requested flight route automatically and safely. The requested flight route is shown with red lines in Fig. 2.

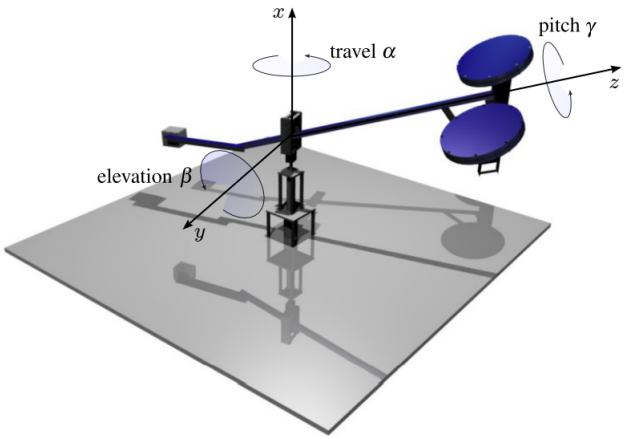

Fig. 1. CAD-Model of QUANSER 3-DOF helicopter

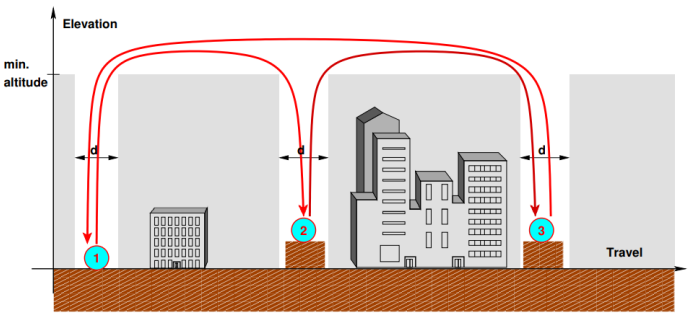

Fig. 2. Requested flight route

The helicopter should be able to do the following jobs without flying through the gray shaded regions:

- Start at point 1 and keep the elevation.
- Fly to point 2 and pick up the cargo.
- Fly to point 3 and deposit the cargo.
- Fly back to starting point 1 and land safely.

The following pages will show how we identify the black box model, modelling the plant, design the controller and finish the cargo transport mission, according to the systematic approach guideline.

1.2 Protocol Structure

In Chapter 2, the simplification of the real plant is discussed in detail. And the kinetic equations are established for the further system analysis and controller design.

In Chapter 3, a time-invariant state space expression of the linearized system is defined. The detailed design processes for LQI controller and Luenberger observer are introduced.

In Chapter 4, the LQI controller and an observer that are introduced in the last chapter are applied to the self-defined nonlinear model, so that the controller design can be verified and further it can be applied to the black box model.

In Chapter 5, the black box model is controlled by this LQI controller to track a reasonably defined trajectory. The performance and some problems are discussed in detail.

In Chapter 6, the performance of the real plant with re-tuned LQI controller is shown. The problem regarding to sampling time is discussed roughly.

At the end of this protocol we make a summary of our solution and draw a conclusion of current controller performance.

1.3 Parameters and Variables

In Table 1, we list the important variables of the system and in Table 2, all the important parameters of the plant are defined, which will be used in the equations within this protocol.

Table 1. Variable List.

Variable	Meaning
α	Travel angle
β	Elevation angle
γ	Pitch angle
F_f	Driving force by front motor
F_b	Driving force by back motor
U_f	Voltage of front motor
U_b	Voltage of back motor
U_m	State of magnet

Table 2. Plant Parameter List.

Parameter	Meaning
I_α	Moment of inertia for travel
I_β	Moment of inertia for elevation
I_γ	Moment of inertia for pitch
m_h	Mass of total helicopter
m_{hf}	Mass of helicopter front part
m_{hb}	Mass of helicopter back part
m_{hm}	Mass of one motor
m_{hs}	Mass of one motor shield
m_{hbar}	Mass of helicopter bar
m_{cw}	Mass of counter weight
m_m	Mass of magnet with bar
m_{ma}	Mass of main arm
m_{J2}	Mass of joint 2
m_{eqv}	Unbalanced mass of arms on point m_h
l_{hbar}	Length of helicopter
l_{ma}	Length of main arm
l_{sa}	Length of secondary arm
l_1	Length from joint 1 to 2 along main arm
l_2	Length from m_{cw} to joint 2
e	Length of joint block 1

2. PLANT MODELLING

2.1 Getting Familiar with "Black Box Model"

The first thing we need to do is to get familiar with the given black box model. By interacting with it, we identified the corresponding output ports to three angles and the corresponding input ports to the motors and magnet. They are summarized in the Table 3.

Table 3. Ports of the Black Box Model.

Ports	Meaning
Input 1	Voltage of front motor U_f
Input 2	Voltage of back motor U_b
Input 3	State of magnet U_m
Output 1	Travel angle α
Output 2	Pitch angle γ
Output 3	Elevation angle β

And subsequently, we are able to readout the sensor data, to actuate the helicopter and to control the magnet.

2.2 Motor Characteristics

The formulation of motor characteristics is necessary for controller design. The experimental data of torques under different voltages were provided, and we used this data to perform a second-order interpolation, which gives us the following equations:

$$F_f = \frac{gp_f U_f^2}{1000}$$
$$F_b = \frac{gp_b U_b^2}{1000} \tag{1}$$

with

$$p_f = 6.103\ kg/V^2$$
$$p_b = 4.68\ kg/V^2$$
$$g = 9.81\ N/kg$$
$$U_f,\ U_b \in [0V,\ 4V]$$

2.3 Moment of Inertia for Pitch Calculation

To calculate the moment of inertia for pitch angle γ, the helicopter body is simplified to ease the calculation. The helicopter is simplified to be a bar without mass, on which two point masses m_{hf} and m_{hb} are located at both ends as illustrated in Fig. 3. Each of these two point masses has a half weight of the helicopter.

We can now calculate the moment of inertia for pitch as follows:

$$m_{hf} = m_{hb} = m_{hm} + m_{hs} + \frac{1}{2}m_{hbar} \tag{2}$$

$$I_\gamma = \frac{1}{2}l_{hbar}^2(m_{hf} + m_{hb}) \tag{3}$$

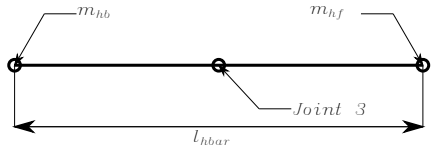

Fig. 3. Helicopter Simplification

2.4 Moment of Inertia for Elevation Calculation

To ease calculate the moment of inertia for elevation angle β, the following simplification process is done.

Firstly, we assume that the main and secondary arms are aligned, and they can be treated as a straight lever with uniformly distributed mass. Secondly, the mass of counter weight m_{cw} is treated as a point mass at one end of the lever. Thirdly, the mass of magnet with bar m_m, the mass of joint 2 m_{J2}, the mass of helicopter m_h and an equivalence mass of this unbalanced lever arm m_{eqv}, are summed up to be one point mass m_1 at the other end of the lever. The simplified model is illustrated in Fig. 4.

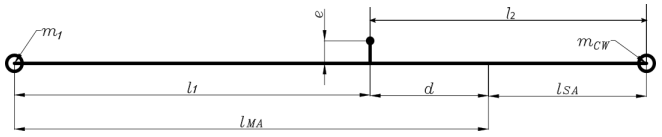

Fig. 4. Model for Elevation

Next, we calculate of the moment of inertia for elevation:

$$I_\beta = m_1(l_1^2 + e^2) + m_{cw}(l_2^2 + e^2) \tag{4}$$

where:

$$m_1 = m_h + m_{J2} + m_m + m_{eqv}$$
$$m_{eqv} = m_{ma}\frac{l_1 - l_2}{l_1 + d}$$

2.5 Moment of Inertia for Travel Calculation

For the calculation of I_α, we use the same simplified model as the model for the calculation of I_β. And the relationship between I_α and I_β can be easily given by:

$$I_\alpha = I_\beta \cos^2(\beta) \tag{5}$$

2.6 Numerical Values of Simplified Model

After inserting the value of plant parameters to Eq. 3 and Eq. 4, we got the numerical values of moments of inertia for elevation I_β and pitch I_γ. The moment of inertia for travel I_α depends on angle β, a reasonable numerical value will be calculated in the following chapter. They are listed in Table 4.

Table 4. Moment of Inertia for Three Axis

Symbols	**Values**
I_γ	$0.0408\ kgm^2$
I_β	$1.1404\ kgm^2$
I_α	$I_\beta \cos^2(\beta)$

2.7 Differential Equations of Motion

We now apply **Euler's second law of motion** to obtain the equations of motions for travel α, elevation β and pitch γ, which are:

$$\begin{aligned} I_\alpha \ddot{\alpha} =& -(F_f + F_b) l_1 \sin\gamma \cos\beta \\ I_\beta \ddot{\beta} =& (F_f + F_b)\cos\gamma l_1 - m_1 g(l_1 \cos\beta + e\sin\beta) \\ &+ m_{cw} g(l_2 \cos\beta - e\sin\beta) \\ I_\gamma \ddot{\gamma} =& (F_f - F_b) b \end{aligned} \tag{6}$$

Rearranging them leads to:

$$\begin{aligned} \ddot{\alpha} =& \frac{-(F_f + F_b)\sin\gamma l_1}{I_\beta \cos\beta} \\ \ddot{\beta} =& ((F_f + F_b)\cos\gamma l_1 - m_1 g(l_1 \cos\beta + e\sin\beta) \\ &+ m_{cw} g(l_2 \cos\beta - e\sin\beta))/I_\beta \\ \ddot{\gamma} =& \frac{(F_f - F_b) b}{I_\gamma} \end{aligned} \tag{7}$$

3. MODEL ANALYSIS AND CONTROLLER DESIGN

3.1 Linearization

We choose $x = (\alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma})^T$ to be the system state, and choose $u = (F_f, F_b)^T$ as system input, where F_f and F_b are the generated forces of the front and back motor, respectively. Then, the dynamics of the helicopter can be described by the following general differential equation:

$$\dot{x} = f(x, u) \tag{8}$$

Ignoring the higher derivative terms, we can get the linearized system around one working point $(x_0,\ u_0)$ with a state-space form

$$\Delta\dot{x} = \underbrace{\frac{\partial f(x,u)}{\partial x}|_{x=x_0,u=u_0}}_{A} \Delta x + \underbrace{\frac{\partial f(x,u)}{\partial u}|_{x=x_0,u=u_0}}_{B} \Delta u \tag{9}$$

where

$$\Delta x = x - x_0$$
$$\Delta u = u - u_0$$

To find out the matrix A and B, we need to calculate explicitly with a desired linearization point. The helicopter elevates from $-27°$ to $-5°$. If we choose a higher linearization point, the performance of travel angle α will be better, but the overshoot of elevation angle β at the beginning will be much bigger. Similarly, if we choose a lower linearization point, the performance of travel angle α will be worse, but the overshoot of elevation angle β at the beginning will be much smaller. Therefore, we choose the approximate middle point of this range to be the linearization point.

$$x_0 = (0, 0, \frac{-15\pi}{180}, 0, 0, 0)^T$$
$$u_0 = (F_{f0},\ F_{b0})^T$$

Then, u_0 can be calculated by the following equations:

$$\begin{aligned} F_{f0} + F_{b0} =& \frac{1}{l_1}(m_1 g(l_1 \cos\beta_0 + e\sin\beta_0) \\ &- m_{CW} g(l_2 \cos\beta_0 - e\sin\beta_0)) \\ F_{f0} - F_{b0} =& 0 \end{aligned} \tag{10}$$

where $\beta_0 = \frac{-15\pi}{180}$ is chosen. The numerical result is:

$$F_{f0} = F_{b0} = 0.3411N$$

The analytical result for matrix A can be derived element-wise, obviously many of the terms are zero. The analytical results of these non-zero terms around the linearization point are:

$$\begin{aligned} \frac{\partial\ddot{\alpha}}{\partial\beta} =& \frac{(F_{f0} + F_{b0})\sin\gamma_0 \sin\beta_0 l_1}{I_\beta \cos^2\beta_0} \\ \frac{\partial\ddot{\alpha}}{\partial\gamma} =& \frac{-(F_{f0} + F_{b0})\cos\gamma_0 l_1}{I_\beta \cos\beta_0} \\ \frac{\partial\ddot{\beta}}{\partial\beta} =& \frac{1}{I_\beta}(m_1 g(l_1 \sin\beta_0 - e\cos\beta_0)) \\ &- \frac{1}{I_\beta}(m_{CW} g(l_2 \sin\beta_0 + e\cos\beta_0)) \\ \frac{\partial\ddot{\beta}}{\partial\gamma} =& \frac{(F_f + F_b)\sin\gamma_0 l_1}{I_\beta} \end{aligned} \tag{11}$$

Similarly for the input matrix B, the non-zero terms around the linearization point are:

$$\begin{aligned}\frac{\partial\ddot{\alpha}}{\partial F_b} &= \frac{\partial\ddot{\alpha}}{\partial F_f} = \frac{-\sin\gamma_0 l_1}{I_\beta \cos\beta_0}\\ \frac{\partial\ddot{\beta}}{\partial F_b} &= \frac{\partial\ddot{\beta}}{\partial F_f} = \frac{\cos\gamma_0 l_1}{I_\beta}\\ \frac{\partial\ddot{\gamma}}{\partial F_f} &= -\frac{\partial\ddot{\gamma}}{\partial F_b} = \frac{b}{I_\gamma}\end{aligned} \tag{12}$$

Hence, the linearized state space expression of the system can be written as:

$$\begin{aligned}\Delta\dot{x} &= A\Delta x + B\Delta u\\ \Delta y &= C\Delta x\end{aligned} \tag{13}$$

with the matrices:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & \frac{\partial\ddot{\alpha}}{\partial\beta} & 0 & \frac{\partial\ddot{\alpha}}{\partial\gamma} & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & \frac{\partial\ddot{\beta}}{\partial\beta} & 0 & \frac{\partial\ddot{\beta}}{\partial\gamma} & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0\\ \frac{\partial\ddot{\alpha}}{\partial F_f} & \frac{\partial\ddot{\alpha}}{\partial F_b}\\ 0 & 0\\ \frac{\partial\ddot{\beta}}{\partial F_f} & \frac{\partial\ddot{\beta}}{\partial F_b}\\ 0 & 0\\ \frac{\partial\ddot{\gamma}}{\partial F_f} & \frac{\partial\ddot{\gamma}}{\partial F_b} \end{pmatrix} \tag{14}$$

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

3.2 Controllability and observability analysis

The analytical solution of the system matrix A, input matrix B and output matrix C are given in (14). To analyze the controllability and observability of the system, the numerical value of the matrices needs to be calculated. Applied the chosen linearization point x_0 and corresponding input u_0, the following numerical values can be obtained:

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & -0.4056 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & -1.409 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\\ B &= \begin{pmatrix} 0 & 0\\ 0 & 0\\ 0 & 0\\ 0.5744 & 0.5744\\ 0 & 0\\ 4.458 & -4.458 \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned} \tag{15}$$

Next, we compute the numerical value of the controllability Matrix P:

$$\begin{aligned} P &= (B\ AB\ A^2B\ A^3B\ A^4B\ A^5B)\\ &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & ...\\ 0 & 0 & 0 & 0 & -1.8082 & ...\\ 0 & 0 & 0.5744 & 0.5744 & 0 & ...\\ 0.5744 & 0.5744 & 0 & 0 & -0.8093 & ...\\ 0 & 0 & 4.458 & -4.458 & 0 & ...\\ 4.458 & -4.458 & 0 & 0 & 0 & ... \end{pmatrix} \end{aligned} \tag{16}$$

P has the rank of 6, which means $rank(P) = n = 6$, and the system will be fully controllable at the linearization point (x_0, u_0).
An observer is necessary for a full state feedback control, we proceed to obtain the numerical value of the observability Matrix M:

$$\begin{aligned} M^T &= (C^T\ A^TC^T\ A^{2^T}C^T\ A^{3^T}C^T\ A^{4^T}C^T\ A^{5^T}C^T)\\ &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & ...\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & ...\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1.409 & ...\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & ...\\ 0 & 0 & 1 & 0 & 0 & 0 & -0.4056 & 0 & ...\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & ... \end{pmatrix} \end{aligned} \tag{17}$$

M has the rank of 6, which means $rank(M) = n = 6$, and the system will be fully observable at the linearization point (x_0, u_0).

3.3 Requirements of the Controller

Taking the targets and the constraints of the mission into consideration, the controller must fulfill the following requirements.

- the input to the system given by the controller should be in the range of $0V$ to $4V$.
- the controller should be able to stabilize pitch angle γ at any time.
- For α and β, there should be no steady errors, which means the real plant should be able to track the desired trajectory fast enough.

3.4 Controller Design

After a few iterations, we decide to use a LQI controller with a static pre-filter to finish the task. A LQI controller can make sure that the closed loop system is stabilized. A static pre-filter is used to provide a quicker response.

The block diagram of our system is as follows:

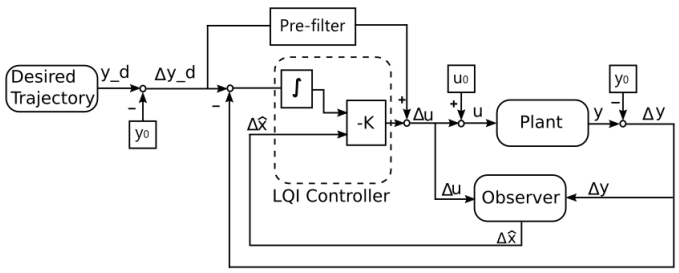

Fig. 5. Control Block Diagram

Δu can be expressed as:

$$\begin{aligned} \Delta u &= V\Delta y_d - K\Delta x - K_I\int_0^t (\Delta y_d(\tau) - \Delta y(\tau))d\tau\\ &= V\Delta y_d - K_{lqi}z\\ &= V\Delta y_d + \Delta u_{lqi} \end{aligned} \tag{18}$$

where:

$$\begin{aligned} V &= -(C(A-BK)^{-1}B)^{-1}\\ z &= (\ \Delta x \quad \int_0^t (\Delta y_d(\tau) - \Delta y(\tau))d\tau\ )^T = (\ \Delta x \quad x'\ )^T\\ K_{lqi} &= (\ K \quad K_I\ ) \end{aligned}$$

The LQI controller is a minimizer of the following optimization problem:

$$J(\Delta u, z) = \int_0^T (z^T Q z + \Delta u^T R \Delta u + 2 z^T N \Delta u) dt \quad (19)$$

$$\begin{aligned} s.t. \quad & \Delta \dot{x} = A\Delta x + B\Delta u \\ & \Delta y = C\Delta x \\ & \Delta u = K_{lqi}\Delta x \\ & z = (\ \Delta x \quad \int_0^t (\Delta y_d(\tau) - \Delta y(\tau)) d\tau\ )^T \end{aligned}$$

Finally, the conversion from linear system input Δu to plant voltage inputs U_f, U_b must be performed before it is given to the plant. From (1) and (9) we have:

$$\begin{aligned} (\ F_f \quad F_b\ )^T &= u_0 + \Delta u \\ U_i &= \sqrt{\frac{1000|F_i|}{g p_i}} \qquad i \in \{f,\ b\} \end{aligned} \quad (20)$$

3.5 Observer Design

The implementation of LQI controller needs an observer to get the full state of the system. The linear system (13) is used again for observer design.

The observer contains two terms, the simulation term and the correction term, it can be directly written as:

$$\Delta \dot{\hat{x}} = A\Delta\hat{x} + B\Delta u + E(\Delta y - C\Delta\hat{x}) \quad (21)$$

The error matrix $(A - EC)$ of the observer should have poles at left side of the complex plane, and the observer should be at least three times faster than the closed loop to deliver a good performance. We set the eigenvalues of error matrix $(A - EC)$ as follows:

$$eig(A - EC) = \begin{pmatrix} -23 \\ -24 \\ -25 \\ -26 \\ -27 \\ -28 \end{pmatrix}$$

We check the system responce by calculate the eigenvalues of matrix $(A - BK)$:

$$eig(A - BK) = \begin{pmatrix} -6.0885 + 0.0000i \\ -3.3665 + 0.0000i \\ -0.9426 + 2.2804i \\ -0.9426 - 2.2804i \\ -2.8339 + 3.0183i \\ -2.8339 - 3.0183i \end{pmatrix}$$

Thus, we can now compute the matrix E using the numerical values of matrices A, B and C in (15)

$$E = \begin{pmatrix} 53.1961 & 0.925 & -0.0001 \\ 706.6368 & 24.4921 & -0.4076 \\ 1.0361 & 52.8039 & -0.0001 \\ 27.5054 & 694.9511 & -0.0022 \\ 0 & 0 & 47 \\ -0.0003 & -0.0003 & 552 \end{pmatrix}$$

4. WORK WITH NONLINEAR MODEL

Now we are ready to tune a LQI controller for our self-defined nonlinear plant. The details will be showed in this chapter.

4.1 Numerical Values of the Controller

The matrix K_{lqi} can be calculated by Matlab with the command

$$[K, S, e] = lqi(SYS, Q, R, N)$$

.

When using the Matlab lqi() command, the weighting matrices $Q \in \mathbf{R}^{8\times8}$, $R \in \mathbf{R}^{2\times2}$ and $N \in \mathbf{R}^{2\times8}$ of augmented state z and linear system input Δu should be given as arguments.

The weighting matrices are chosen iteratively. The last two diagonal elements of the weight matrix Q are chosen as big values, because they correspond to the integration of the error between the desired and the real travel and elevation trajectory, which we previously defined as x'. After several trials, we choose the matrices as follows:

$$\begin{aligned} Q &= \begin{pmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 \end{pmatrix} \quad R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ N &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T \end{aligned} \quad (22)$$

The resulting gain matrix K is:

$$K = \begin{pmatrix} -12.7320 & 12.7320 \\ -15.7104 & 15.7104 \\ 4.7941 & 4.7941 \\ 2.9744 & 2.9744 \\ 4.4030 & -4.4030 \\ 1.2195 & -1.2195 \\ 5.0000 & -5.0000 \\ -5.0000 & -5.0000 \end{pmatrix}^T \quad (23)$$

4.2 Performance with Self-defined Nonlinear Model

To check the performance of the LQI controller with our self-defined nonlinear model, polynomial trajectories are defined for the travel and the elevation.

Travel angle α should finish one full round from $t = 10s$ to $t_f = 35s$. The desired and real trajectories are shown in Fig. 6.

Elevation angle β should go from $\beta = -27°$ to $-5°$ within 10 seconds. After 25 seconds it goes down to $\beta = -22°$. The desired and real trajectories are shown in Fig. 7.

Fig. 6 and Fig. 7 show that the controller can stabilize our self-defined nonlinear model very well and can make it converge to a reasonably defined smooth trajectory

quickly. For both of travel angle α and elevation angle β, there are no steady state errors between desired and real trajectories. Therefore we can draw a conclusion that our controller has the ability to solve the task with our self-defined nonlinear model. Then, we can go further to implement the controller to the black box model, which will be explained in next chapter.

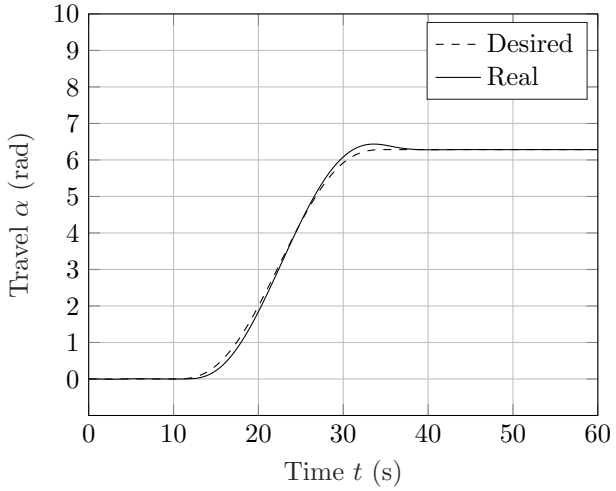

Fig. 6. Desired and Real Travel Trajectory

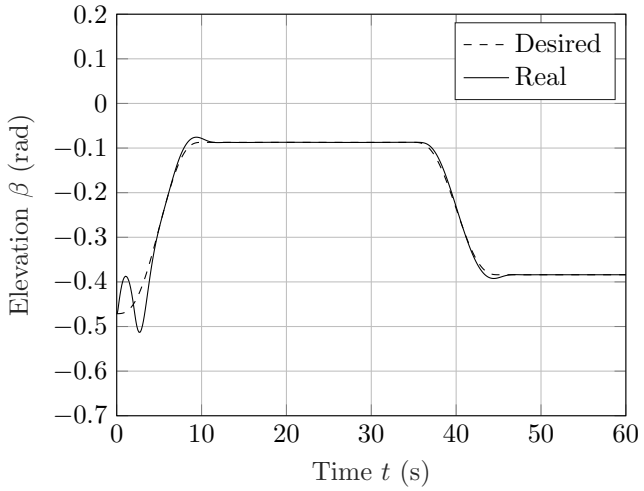

Fig. 7. Desired and Real Elevation Trajectory

5. WORK WITH BLACK BOX MODEL

5.1 Desired Trajectory

We design all parts of our desired trajectory as polynomial functions with respect to time, which is the same method used for the self-defined nonlinear model. The polynomial functions have the highest order of five. By choosing the right coefficients, it is possible to make sure that not only the initial and final velocity but also the initial and final acceleration in each step are zero. A general expression of such a trajectory is:

$$\begin{aligned} r(t) &= r_0 + (r_1 - r_0)\sum_{i=3}^{5} p_i \left(\frac{t - t_0}{t_1 - t_0}\right)^i \\ (\ p_3 \ \ p_4 \ \ p_5\ ) &= (\ 6 \ \ -15 \ \ 10\ ) \\ t &\in [t_0,\ t_1] \end{aligned} \tag{24}$$

The whole process is divided into 11 steps as follows:

Table 5. Desired Trajectory.

Step	Time	Action
1	0–5s	β goes from $-27°$ to $-5°$
2	5–15s	α goes from $0°$ to $90°$
3	15–23s	β goes from $-5°$ to $-22°$
4	23–27s	No movement, pick up cargo
5	27–32s	β goes from $-22°$ to $-5°$
6	32–52s	α goes from $90°$ to $450°$
7	52–60s	β goes from $-5°$ to $-22°$
8	60–63s	No movement, deposit cargo
9	63–68s	β goes from $-22°$ to $-5°$
10	68–88s	α goes from $450°$ to $0°$
11	88–96s	β goes from $-5°$ to $-27°$

5.2 Performance with Black Box Model

After applying the desired trajectory to our system, we have to re-tune the state weighting matrix Q to be Q_r. Matrices R and N stay unchanged:

$$Q_r = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1000&0 \\ 0&0&0&0&0&0&0&1000 \end{pmatrix} \tag{25}$$

Tuning the weight matrix Q_r is an iterative process. We realize that the most important elements in the weight matrix Q_r for our controller are the last two diagonal elements, which correspond to the integration of the error between the desired and real trajectory. Therefore they are given with large values.

The resulting gain matrices K is re-calculated to be K_r:

$$K_r = \begin{pmatrix} -34.4904 & 34.4904 \\ -26.5888 & 26.5888 \\ 15.8786 & 15.8786 \\ 5.3052 & 5.3052 \\ 4.9789 & -4.9789 \\ 1.2714 & -1.2714 \\ 22.3607 & -22.3607 \\ -27.3861 & -27.3861 \end{pmatrix}^T \tag{26}$$

Applying K_r to the controller, we got a good overall performance in tracking the desired trajectories as shown in the following figures.

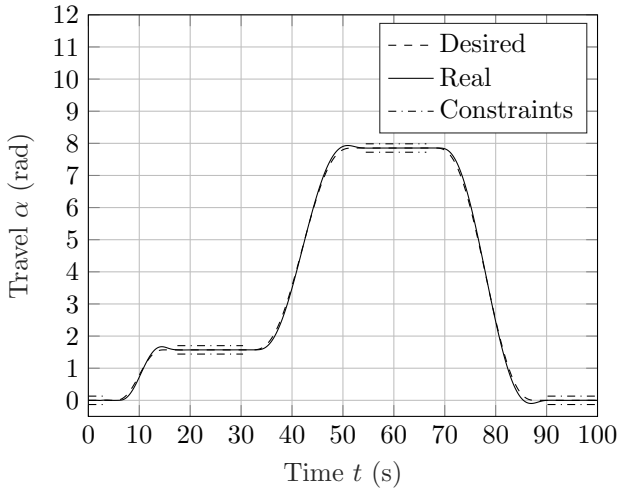

Fig. 8. Desired and Real Travel Trajectory

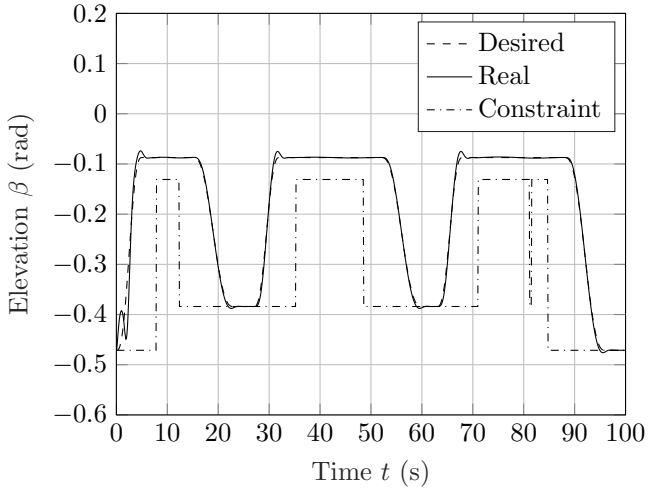

Fig. 9. Desired and Real Elevation Trajectory

Fig. 8 and Fig. 9 show that our new LQI controller can stabilize the black box model and track the desired trajectories very well without violating the constraints.

But there is a "swinging" behaviour as shown in Fig. 10, which gives a more detailed look of elevation β from $t = 0s$ to $t = 25s$. The main reason behind is the difference between the linearization point and initial point. As described in Sec. 3.1, we chose $(0, 0, \frac{-15\pi}{180}, 0, 0, 0)^T$ as our linearization point, but the system begins at point $(0, 0, \frac{-27\pi}{180}, 0, 0, 0)^T$. This large gap in elevation angle β results in a sudden large feedback, trying to drive the model back to $\beta_0 = \frac{-15\pi}{180}$. Within a short period, the contribution of integrator is getting larger, suppressing the input voltage to be much smaller than that of beginning. Therefore, the real trajectory starts to fall behind to the desired one. This can be interpreted as a compensation of the error integration. As a result, the whole process shows a swinging behaviour. That requires us to choose a reasonable linearzation point, to compromise between a good performance of elevation behavior and a good system linear approximation. $\beta_0 = \frac{-15\pi}{180}$ is a good choice for us.

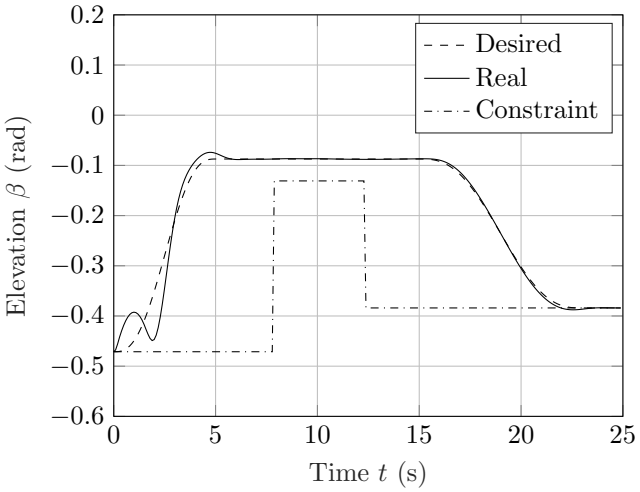

Fig. 10. Swinging Behaviour and Undershoot of Elevation

At last, to pick up the cargo and deposit it in a proper time, a simple control signal for the magnet is designed as shown in Fig. 11.

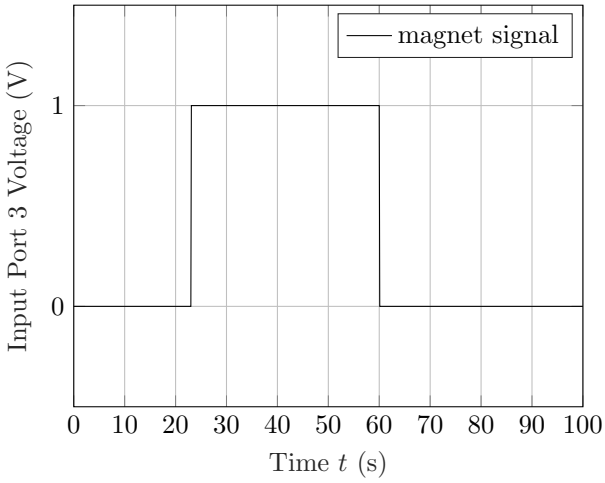

Fig. 11. Magnet Control Signal

6. WORK WITH REAL PLANT

6.1 Performance with Real Plant

During the final Lab-work, the controller and observer should be able to interact with the real plant, and finish the transport task.

After building the connection between Matlab and the QUANSER hardware by using Hardware-in-the-loop Simulation and configuring the settings of Matlab properly, the sensor data can be read by the scope in Simulink. Being applied with the controller, which had the same weighting matrices that are defined in Chapter 5, the real plant was unstable. After several trials, the sampling time was set to be $dt = 0.005s$, and the weighting matrix Q to be Q_{lab} shown in Eq. 27. Then the real plant was stabilized and could follow the desired trajectory.

The performance of the real plant is shown from Fig. 12 to Fig. 15.

$$Q_{lab} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 \end{pmatrix} \tag{27}$$

From Fig. 12, it can be observed that the helicopter could follow up with the desired trajectory. And it was really satisfying that during the ascending and descending phases, the helicopter could stay inside the corridors most of the time, except for the entering.

From Fig. 13, it can be seen that the helicopter can also track the elevation trajectory without steady error, which was great. But an violation occurred again when it landed.

The reason for the constraint violations during entering corridors is because our desired trajectory is static. That means once it is generated and given to the plant, it cannot be changed and cannot be updated during run-time.

Fig. 14 and Fig. 15 show that the magnet was switched on and off in the right time and the pitch angle was always be controlled to be in a reasonable range.

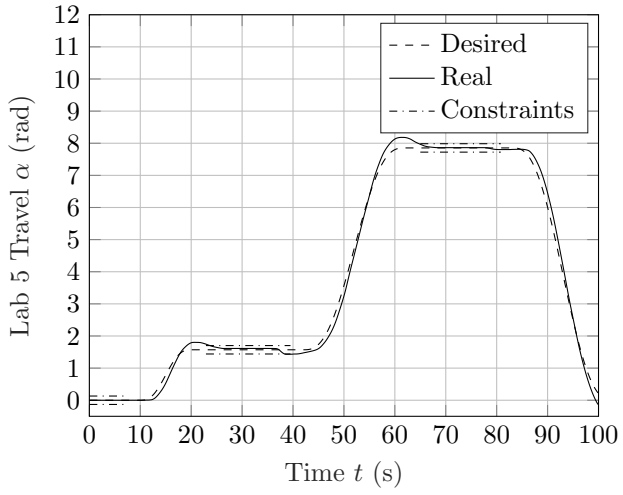

Fig. 12. Desired and Real Travel Trajectory in Lab 5

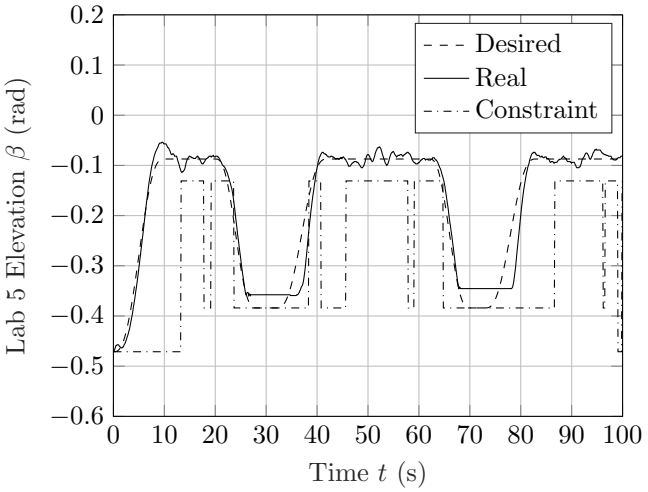

Fig. 13. Desired and Real Elevation Trajectory in Lab 5

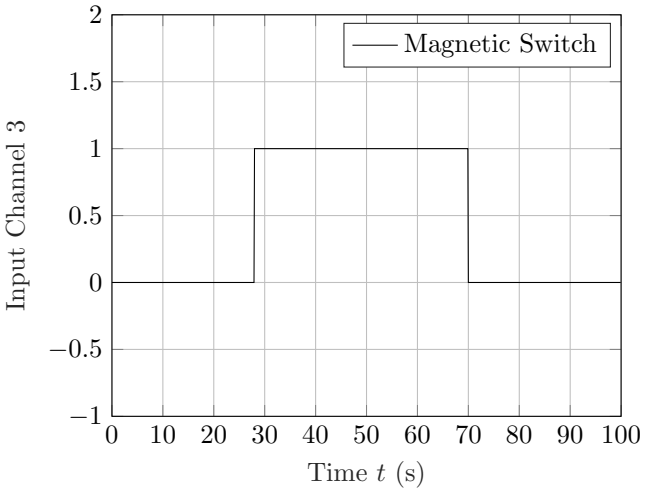

Fig. 14. Magnet Control Signal in Lab 5

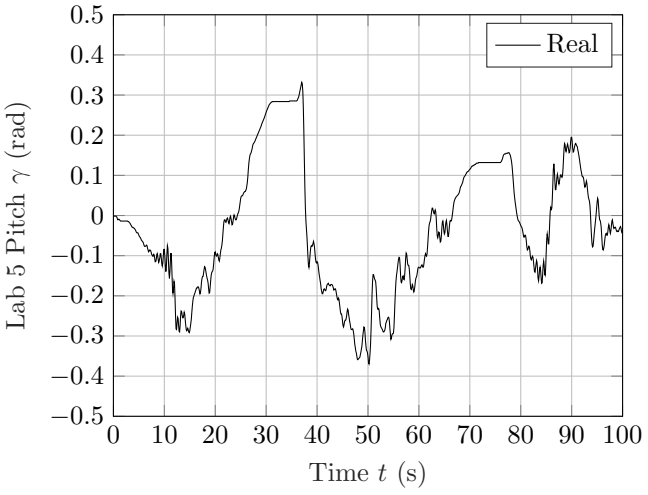

Fig. 15. Real Pitch in Lab 5

6.2 Discussion about Sampling Time

During the final Lab-work, the weighting matrices did not have much contribution to the stability of the real plant, whereas different sampling times will affect the performance greatly.

We think it may have something to do with the differential parts $\Delta\dot{\hat{\alpha}}$, $\Delta\dot{\hat{\beta}}$ and $\Delta\dot{\hat{\gamma}}$ in the estimated state $\Delta\hat{x}$, which may be greatly influenced by the sampling time for discrete system.

As is expressed in Eq. 18, the increment of input Δu can be written in the following form:

$$\Delta u = V\Delta y_d - K\Delta x - K_I \int_0^t (\Delta y_d(\tau) - \Delta y(\tau))d\tau$$

We can rewrite this expression into a discrete form with arguments explicitly written as follows:

$$\begin{aligned}\Delta u(t_k, dt) =& V\Delta y_d(t_k) - K\Delta x(t_k, dt) \\ &- K_I \sum_{t=0}^{kdt} (\Delta y_d(t) - \Delta y(t))\end{aligned} \tag{28}$$

where:

$$\Delta x(t_k, dt) = \begin{pmatrix} \Delta\alpha(t_k) \\ \dfrac{\alpha(t_k) - \alpha(t_k - t_{k-1})}{dt} \\ \Delta\beta(t_k) \\ \dfrac{\beta(t_k) - \beta(t_k - t_{k-1})}{dt} \\ \Delta\gamma(t_k) \\ \dfrac{\gamma(t_k) - \gamma(t_k - t_{k-1})}{dt} \end{pmatrix} \tag{29}$$

$$\Delta y(t_k) = C\Delta x(t_k, dt) = \begin{pmatrix} \Delta\alpha(t_k) \\ \Delta\beta(t_k) \\ \Delta\gamma(t_k) \end{pmatrix} \tag{30}$$

In the real implementations, Δx is estimated by a Luenberger observer to be $\Delta\hat{x}$. But our Luenberger observer is only designed for a continous system, we didn't consider the effect of sampling time, and we cannot guarantee the observer will still be stable.

Intuitively speaking, the observer wants to correct the estimation error of $\Delta\dot{\hat{\alpha}}(t_k, dt)$, $\Delta\dot{\hat{\beta}}(t_k, dt)$ and $\Delta\dot{\hat{\gamma}}(t_k, dt)$ in $\Delta\hat{x}(t_k, dt)$ by using the real output $\Delta y(t_k)$ and estimated output $\Delta\hat{y}(t_k)$, but both of them do not carry any information about the sampling time dt. Therefore, the estimation of these three states may not be accurate.

7. CONCLUSION

Following the systematic approach for solving control problems, we modeled the plant with nonlinear differential equations and linearized them to a linear time-invariant system, which is expressed in a state space. After this, a LQI controller with pre-filter and a Luenberger observer are designed to make sure that our helicopter can have a good performance in tracking the desired trajectory.

The real plant can also be stabilized and have a relatively good behaviour in tracking the desired trajectory, except for some inaccuracy in landing and avoiding the collisions to constraints. But these inaccuracy can be further improved by tuning the weighting matrices and changing the trajectory generator. Besides, a further investigation about effects of sampling time to the observer and the controller should be done, and this will be helpful to the understanding of discrete-time system stability.

We can draw a conclusion that, our LQI controller with pre-filter and Luenberger observer can basically finish the task.

REFERENCES

[1] Lunze, J. (2020). Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. Available online from `https://link.springer.com/book/10.1007/978-3-662-60746-6`

[2] MathWorks, Inc. (2020). Control System Toolbox User's Guide (R2020b). Available online from `https://de.mathworks.com/help/pdf_doc/control/control_ug.pdf/`

[3] "3-DOF Helicopter: User Manual" Quanser Inc., Markham, ON, Canada., Tech. Rep., 2010.

[4] "Q8-USB Data Acquisition Board: User Manual" Quanser Inc., Markham, ON, Canada., Tech. Rep., 2010.

[5] "Quanser QUARC: Short Introduction" IST, Tech. Rep., 2011.

[6] "CAD drawings of the Quanser 3 DOF helicopter" IST, Tech. Rep., 2011.

[7] "How to write a protocol for the MSc laboratory course "Concepts of Automatic Control" at the IST" IST, Tech. Rep., 2011.