TTK4135 Optimization and Control
Helicopter Lab Report

Group 56

502425
502423
502427

NTNU

Department of Engineering Cybernetics
Norwegian University of Science and Technology

March 14, 2019

Contents

1 Introduction

The optimization concept is implemented in this project to design the trajectory of the helicopter model, which contains three parts: *optimal control in pitch and travel without feedback*, *optimal control in pitch and travel with feedback* and *optimal control in pitch,travel and elevation with and without feedback*. The model of the helicopter is depicted in Fig. 1. The helicopter driven by two motors consists of a base with an arm attached, which has the helicopter body at one end and a balance weight at the other end. The arm can be moved up and down around an elevation axis (elevation e) and rotate around a vertical axis (travel λ). The helicopter body itself can also rotate around an axis normal to the arm (pitch p).

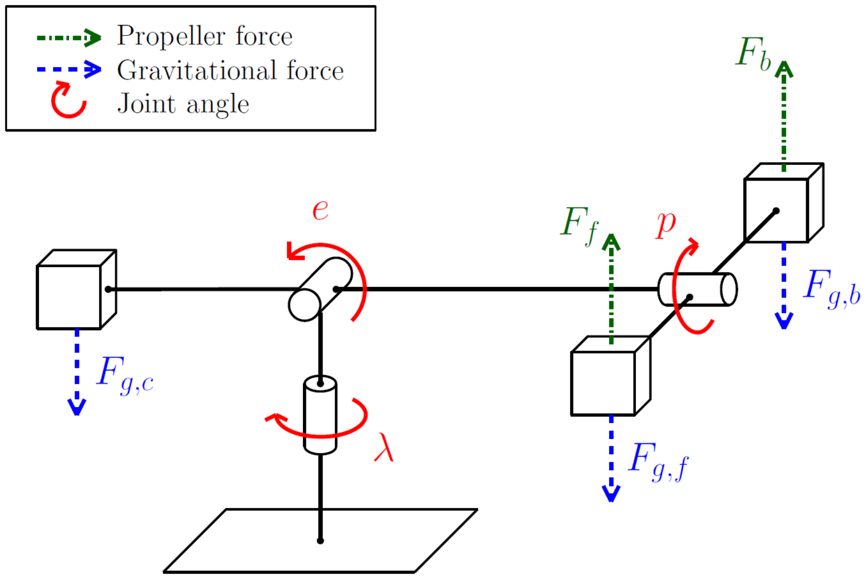

Figure 1: Helicopter model

The optimal path from an initial state to the desired state is computed, both in one and two dimensions. In both cases state feedback is investigated as a way of eliminating drift from the optimal path. Then, a nonlinear constraint is added to the optimization problem. Additional constraints are also investigated as a way of limiting nonlinear effects.

2 Problem Description

In order to analyze, control and optimize the helicopter, we should mathematically model it. the system dynamics about the equilibrium of the helicopter can be described as:

$$
\begin{aligned}
\ddot{p} &= K_1 V_d, \quad K_1 = \frac{K_f l_h}{J_p} \\
\ddot{\lambda} &= -K_2 p, \quad K_2 = \frac{K_p l_a}{J_t} \\
\ddot{e} &= K_3 - \frac{T_g}{J_e}, \quad K_3 = \frac{K_f l_a}{J_e}
\end{aligned}
\tag{2.1}
$$

The following PID controller is implemented to control the height

$$
V_s = K_{ep}(e_c - e) + K_{ei}\int (e_c - e)dt - K_{ed}\dot{e}
\tag{2.2}
$$

and a PD controller is used to control the pitch angle

$$
V_d = K_{pp}(p_c - p) - K_{pd}\dot{p}
\tag{2.3}
$$

We assume that the integral term couteracts from the constant term $-\frac{T_g}{J_e}$, so these two terms can be ignored. Hence the model can be summarized by

$$
\begin{aligned}
\ddot{e} + K_3 K_{ed}\dot{e} + K_3 K_{ep} e &= K_3 K_{ep} e_c \\
\ddot{p} + K_1 K_{pd}\dot{p} + K_1 K_{pp} p &= K_1 K_{pp} p_c \\
\dot{\lambda} &= r \\
\dot{r} &= -K_2 p
\end{aligned}
\tag{2.4}
$$

The main parameters and variables are present below.

Table 1: Parameters and values

Symbol	Parameter	Values	Units
l_a	Distance from elevation axis to helicopter body	0.65	m
l_h	Distance from pitch axis to motor	0.17	m
m_h	Total mass of the motors	0.4	kg
m_g	Effective mass of the helicopter	0.03	kg
J_e	Moment of interia for elevation	0.338	kg·m^2
J_t	Moment of interia for travel	0.338	kg·m^2
J_p	Moment of interia for pitch	0.0116	kg·m^2
K_p	Force to lift the helicopter from the ground	0.2943	N
K_f	Force motor constant	0.0491	N/V

Table 2: Variables

Symbol	Parameter	Units
e	Elevation	rad
$\dot{e}$	Elevation rate	rad/s
e_c	Elevation reference	rad
λ	Travel	rad
r	Travel rate	rad/s
p	Pitch	rad
$\dot{p}$	Pitch rate	rad/s
p_c	Pitch reference	rad

3 Repetition/introduction to Simulink/QuaRC

In this project, MATLAB with Simulink and QuaRC is used for optimization design. So the first step is repetition to the software. The MATLAB software works to create object dynamic models, design and tune feedback loops and supervisory controllers. The simulation models are to validate control designs and automatically generate code for rapid prototyping and production.

4 Optimal Control of Pitch/Travel without Feedback

In this task, given that the elevation is neglected, an optimal trajectory x^* and a corresponding input sequence u^* are computed. In this optimal control, feedback is not included to correct the difference between measured states and optimal ones. This control hierarchy is illustrated in Figure 2.

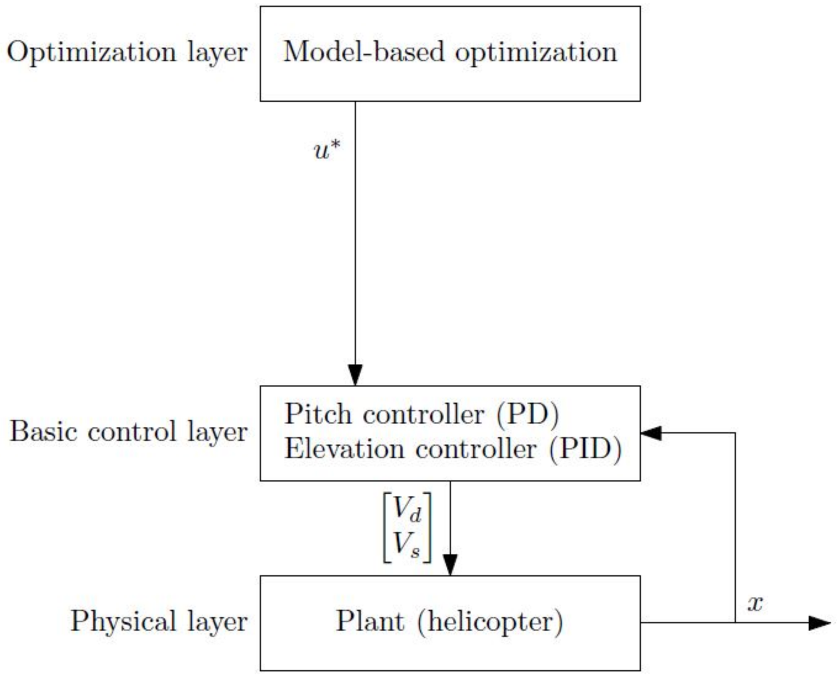

Figure 2: Control hierarchy

4.1 Model on continuous-time state-space form

In the task, the continuous state-space equation is derived in the form as follow,

$$\dot{\mathbf{x}} = \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{u} \tag{4.1}$$

where

$$\mathbf{x} = \begin{bmatrix} \lambda \\ r \\ p \\ \dot{p} \end{bmatrix}, \quad u = p_c \tag{4.2}$$

Based on the model equations in (2.4), equations are rewritten as below,

$$\begin{aligned} \dot{\lambda} &= r \\ \dot{r} &= -K_2 p \\ \ddot{p} &= -K_1K_{pd}\dot{p} - K_1K_{pp}p + K_1K_{pp}p_c \end{aligned} \tag{4.3}$$

Therefore, the continuous state-space equation is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\lambda} \\ \dot{r} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -K_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_1K_{pp} & -K_1K_{pd} \end{bmatrix} \begin{bmatrix} \lambda \\ r \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_1K_{pp} \end{bmatrix} u \tag{4.4}$$

By substituting the numerical values, it is obtained:

$$\mathbf{A}_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.566 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -3.24 & -3.6 \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3.24 \end{bmatrix} \tag{4.5}$$

4.2 Discretization

The system dynamics is implemented as a sequence of constraints in the optimization problem, and must therefore be written in discrete-time state-space form,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \tag{4.6}$$

Here, forward Euler method is used, with a constant time step $\Delta t = 0.25s$:

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\Delta t} \tag{4.7}$$

By substituting (4.7) into (4.1), it results in:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \Delta t \dot{\mathbf{x}}_k \\ &= \mathbf{x}_k + \Delta t(\mathbf{A}_c\mathbf{x}_k + \mathbf{B}_c u_k) \\ &= (\mathbf{I} + \Delta t\mathbf{A}_c)\mathbf{x}_k + (\Delta t\mathbf{B}_c)u_k \\ &= \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k \end{aligned} \tag{4.8}$$

Hence, the discretized state-space equation is

$$\mathbf{x_{k+1}} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & -\Delta t K_2 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & -\Delta t K_1 K_{pp} & 1 - \Delta t K_1 K_{pd} \end{bmatrix} \mathbf{x_k} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \Delta t K_1 K_{pp} \end{bmatrix} u_k \tag{4.9}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0.25 & 0 & 0 \\ 0 & 1 & -0.1415 & 0 \\ 0 & 0 & 1 & 0.25 \\ 0 & 0 & -0.81 & 0.1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.81 \end{bmatrix} \tag{4.10}$$

4.3 Optimal trajectory

Calculate an optimal trajectory for moving the helicopter from $x_0 = [\lambda_0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$ to $x_f = [\lambda_f \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$, where $\lambda_0 = \pi$ and $\lambda_f = 0$. Minimize the following cost function

$$\phi = \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + q p_{ci}^2, \quad q \geq 0 \tag{4.11}$$

We define (4.11) in terms of the full state and input variables:

$$\phi = \frac{1}{2} \sum_{i=0}^{N-1} \{(x_{i+1} - x_f)^T Q (x_{i+1} - x_f) + u_i^T R u_i\} \tag{4.12}$$

where

$$\mathbf{Q} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = 2q \tag{4.13}$$

The system dynamics subject to linear equality constraints are as follow:

$$\underbrace{\left[\begin{array}{cccc|ccc} \mathbf{I} & & & & -\mathbf{B} & & \\ -\mathbf{A} & \ddots & & & & \ddots & \\ & \ddots & \ddots & & & & \ddots \\ & & -\mathbf{A} & \mathbf{I} & & & -\mathbf{B} \end{array}\right]}_{\mathbf{A}_{eq} \in \mathbb{R}^{Nn_x \times N(n_x+n_u)}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\mathbf{z} \in \mathbb{R}^{N(n_x+n_u)\times 1}} = \underbrace{\begin{bmatrix} \mathbf{A}x_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{B}_e q \in \mathbb{R}^{Nn_x \times 1}} \tag{4.14}$$

Matrix $\mathbf{G} \in \mathbb{R}^{N(n_x+n_u)\times N(n_x+n_u)}$ is defined to express (4.12) in terms of the variable $\mathbf{z}$

$$\mathbf{G} = \begin{bmatrix} \mathbf{Q} & & & & & \\ & \ddots & & & & \\ & & \mathbf{Q} & & & \\ & & & R & & \\ & & & & \ddots & \\ & & & & & R \end{bmatrix} \tag{4.15}$$

The QP problem can be stated as

$$\underset{\mathbf{z}}{min} \; \frac{1}{2}\mathbf{z}^T\mathbf{G}\mathbf{z} \tag{4.16}$$

subject to

$$\begin{aligned} \mathbf{A}_{eq}\mathbf{z} &= \mathbf{B}_{eq}, \\ p^{low} &\leq p_k, \; k \in \{1, ..., N\} \\ p_k &\leq p^{high}, \; k \in \{1, ..., N\} \end{aligned} \tag{4.17}$$

with the constraint on the pitch state and input is given by

$$|p| \leq \frac{30\pi}{180}, \; k \in \{1, ..., N\} \tag{4.18}$$

The parameter q weights the relative importance of low input expendature. The plots of variables and input with different q (0.1, 1 and 10) are shown below.

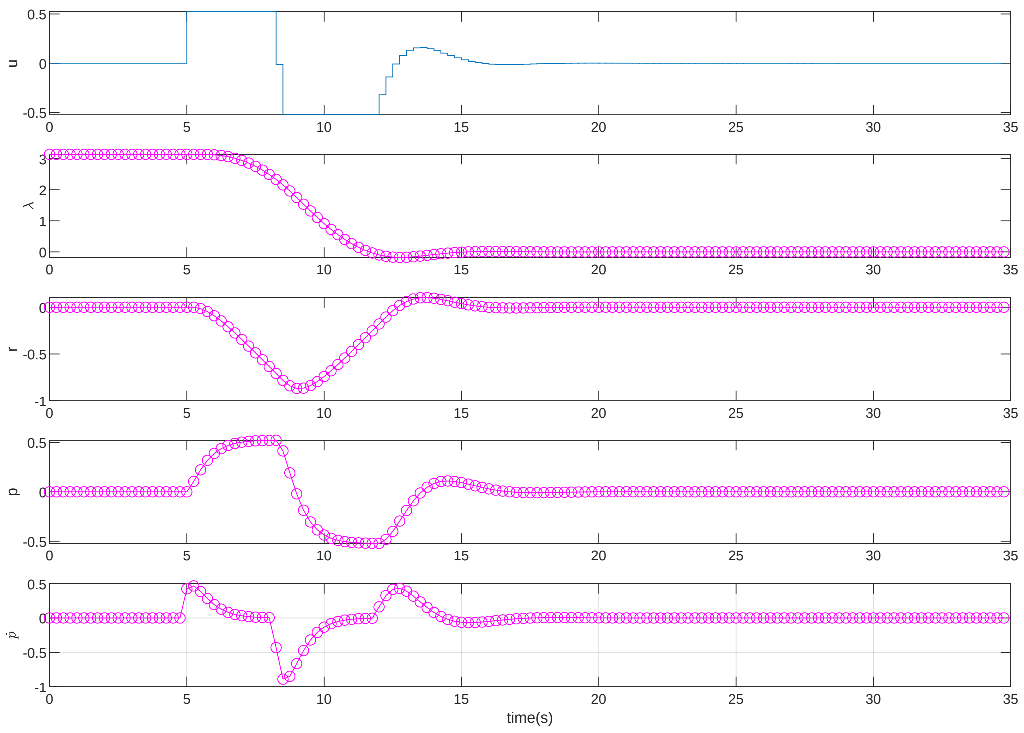

Figure 3: Variables and input with q=0.1

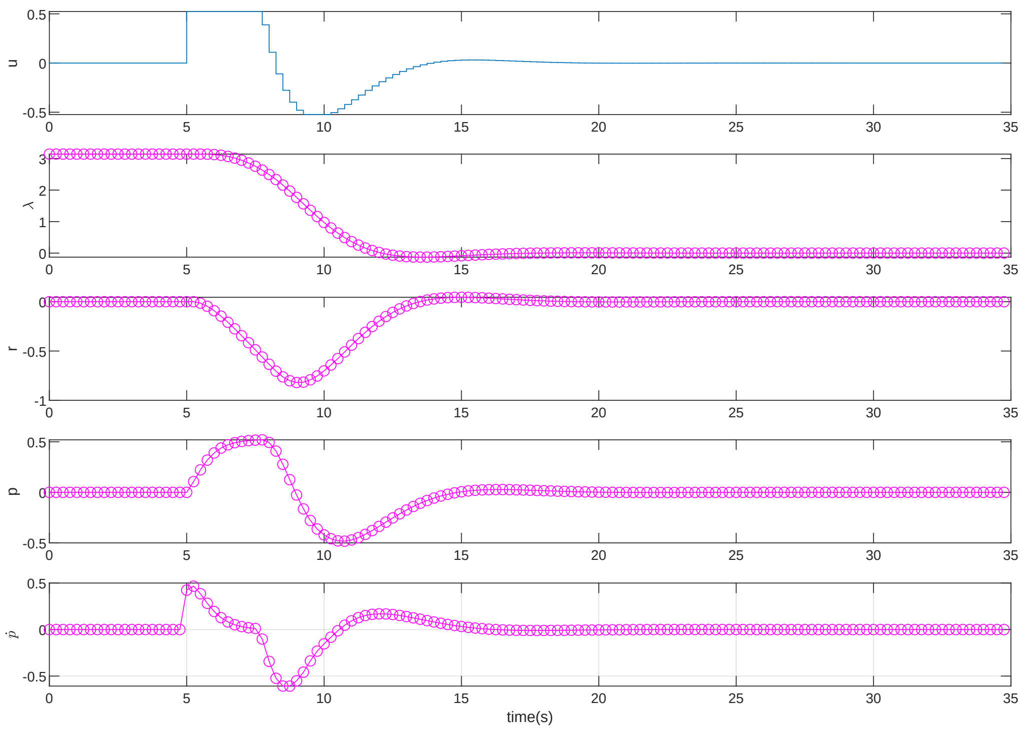

Figure 4: Variables and input with q=1

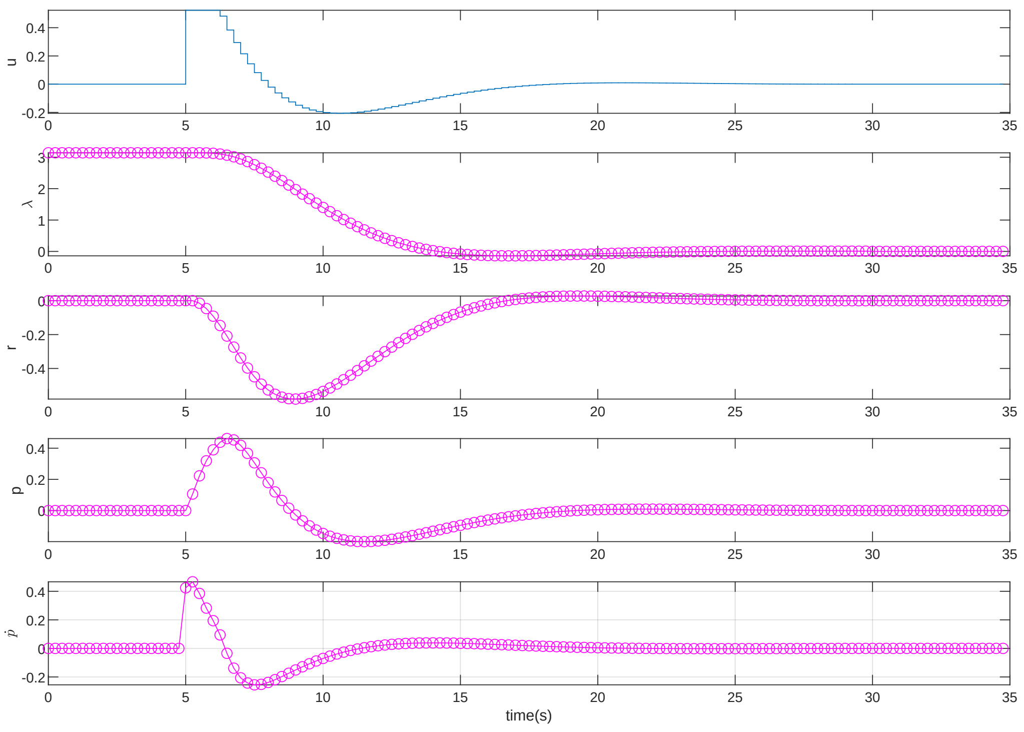

Figure 5: Variables and input with q=10

We see that when q = 0.1 the control is more aggressive and the output settles at desired point faster, since a lower value of q gives a lower penalty on control use. When q = 10 the control is less aggressive and the output converges slowly to desired point since control use is penalized harder.

4.4 Result and Discussion

The optimization control problem is solved by using MATLAB function `quadprog` with $q = 1$. Then the optimal input sequence u^* is implemented to the system. The measured pitch and travel angle of this open loop system are compared with the calculated optimal trajectory x^* in the figures below.

From the comparison, the helicopter is not able to end in the desired point x_f, especially for the travel angle. The reason for this deviation is the open-loop system lacks feedback. Since our assumptions, including neglect of elevation, noise and linearization and discretization error, give a imperfect model, on which the calculation of optimal input sequence is based. This makes the input sequence unfit for the realistic system, and therefore the helicopter cannot end in desired point. To solve this problem, feedback should be introduced to compensate for there errors.

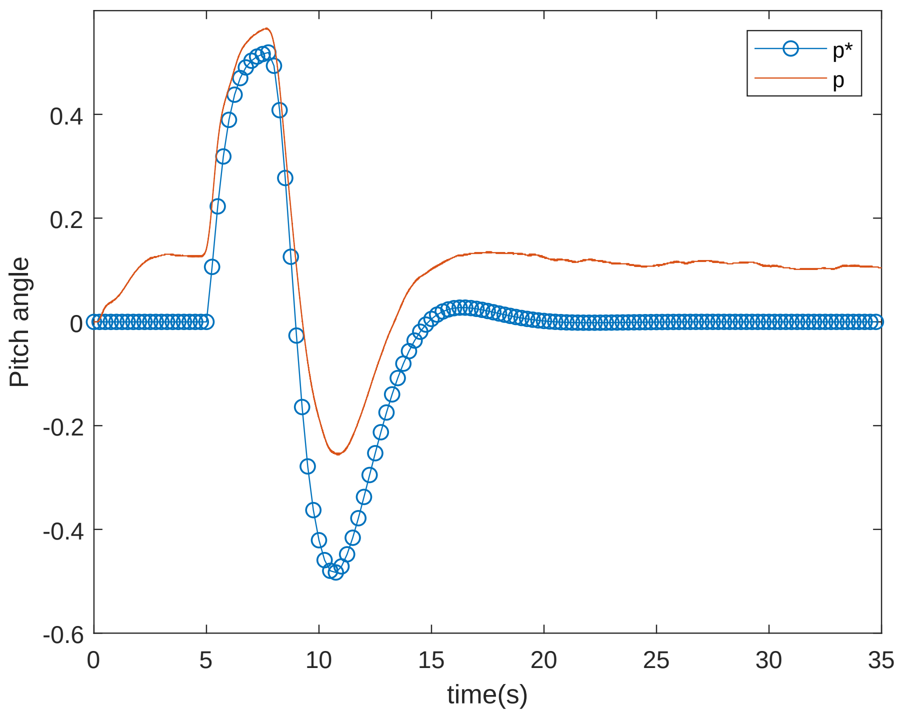

Figure 6: Measured and optimal pitch angle

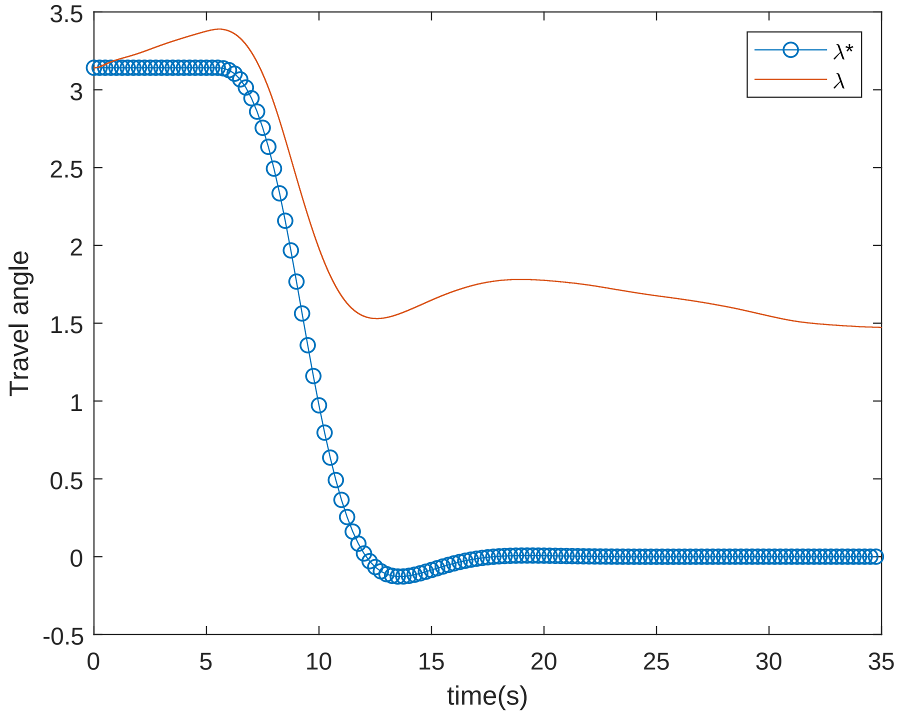

Figure 7: Measured and optimal travel angle

5 Optimal Control of Pitch/Travel with Feedback (LQ)

In this section, a feedback control is implemented to correct deviation between the measurements and control output. The optimal gain of the feedback K is calculated by solving the linear-quadratic regulator problem. What is more, a discussion about advantages and disadvantages between MPC and LQ is done.

5.1 Implementation of feedback

The feedback input is designed as

$$u_k = {u_k}^* - \mathbf{K}^T(\mathbf{x_k} - {\mathbf{x_k}}^*) \tag{5.1}$$

where we define

$$\Delta\mathbf{x}_k = \mathbf{x}_k - {\mathbf{x}_k}^* \tag{5.2}$$

$$\Delta u_k = u_k - {u_k}^* \tag{5.3}$$

The feedback gain is found by minimizing the cost function below:

$$J = \frac{1}{2}\sum_{i=0}^{N-1} {\Delta\mathbf{x}_{i+1}}^T\tilde{\mathbf{Q}}\Delta\mathbf{x}_{i+1} + {\Delta u_i}^T\tilde{\mathbf{R}}\Delta u_i \tag{5.4}$$

Q and **R** matrices are chosen depending on weighting between the cost of state and input. As a consequence, the **P** matrix is the unique positive definite solution of the Riccati equation in solving LQ problem and the **K** matrix is calculated from the formula above.

$$\mathbf{K} = (R + \mathbf{B}^T\mathbf{P}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{P}\mathbf{A} \tag{5.5}$$

5.2 Result and Discussion

In order to have high accuracy in travel, we chose state penalty matrix Q and input penalty matrix R to be

$$\tilde{\mathbf{Q}} = \begin{bmatrix} 8 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{R} = 1 \tag{5.6}$$

The function `dlqr` is used to solve the optimization problem and get the optimal gain K.

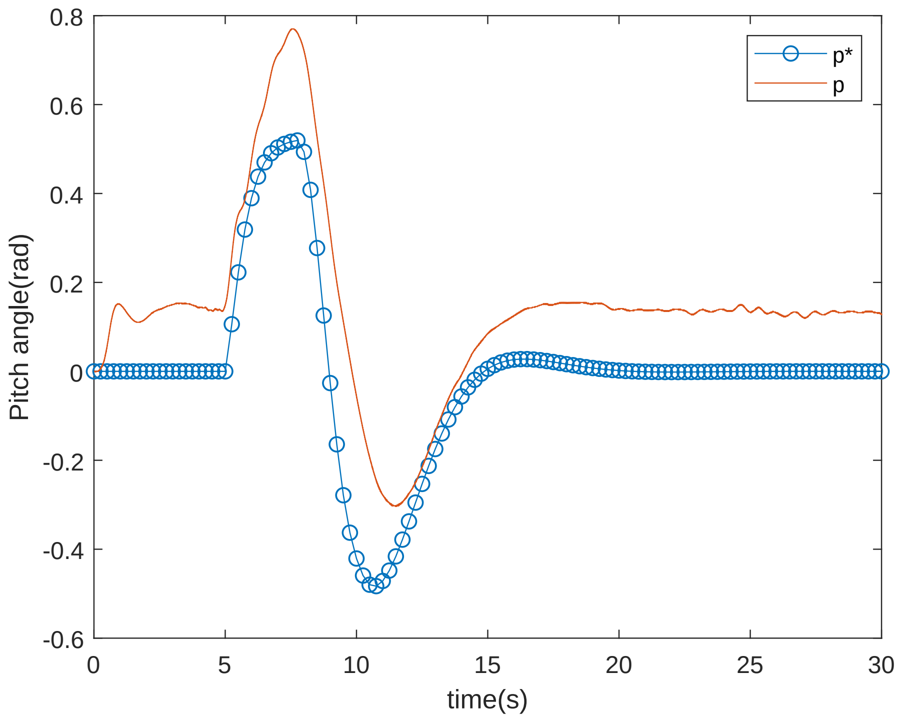

Figure 8: Measured and optimal pitch angle with feedback

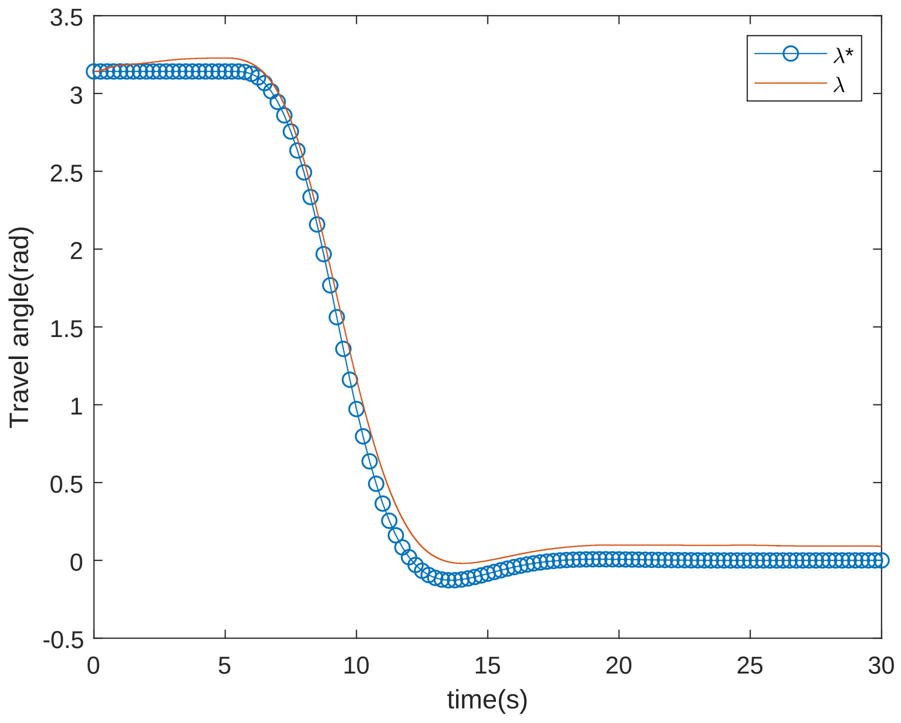

Figure 9: Measured and optimal travel angle with feedback

For the travel, the measured trajectory is very close to the optimal one, only with small deviation at the end.This can be solved by giving more penalty in travel. However, the measured pitch angle deviates from the optimal trajectory obviously. We tried to increase corresponding values in Q matrix in order to penalize the deviation in pitch angle, but the result did not show better performance. We assume this is due to modelling error. Overall, the state feedback controller fulfills the control objective nicely.

5.3 MPC discussion

Model predictive control (MPC) means that the QP problem is solved in each step of optimization and only the first input is used in the next iteration. Since QP is calculated in each step, it seems that the calculated feedback control is more precise while the disadvantage is also obvious—in each iteration, more time is required for optimization. However, the LQ controller only solves the QP problem once and use the calculation result in the following steps, which may lead to inexact result. Therefore, MPC should be recommended when disturbance and noise are large, which makes it possible to recalculate the optimal path to solve rapid changing condition while the helicopter model requires fast dynamics and MPC cannot perform rapidly compared to LQ. So LQ is recommended in this project. The control structure with MPC would substitute the "Advanced control layer" with the "Optimization layer", leading to one less layer in the LQ structure.

6 Optimal Control of Pitch/Travel and Elevation with and without Feedback

In this section, the elevation and elevation rate is considered in the model to calculate the optimal trajectory for the helicopter. The trajectory is assigned to move from the initial point $x_0 = [\pi \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$ to the reference point $x_f = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$ under required constraints. Since there exist nonlinear constraints, a new function `fnoncon` is implemented to solve nonlinear problem.

6.1 Model on continuous-time state-space form

By introducing elevation and elevation rate, the state-space equation is extended.

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -K_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -K_1K_{pp} & -K_1K_{pd} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -K_3K_{ep} & -K_3K_{ed} \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ K_1K_{pp} & 0 \\ 0 & 0 \\ 0 & K_{ep} \end{bmatrix} \mathbf{u} \tag{6.1}$$

where $\mathbf{x} = [\lambda \quad \dot{\lambda} \quad p \quad \dot{p} \quad e \quad \dot{e}]^{\mathrm{T}}$ and $\mathbf{u} = [p_c \quad e_c]^{\mathrm{T}}$.

We note that the first four states are completely decoupled from the last two. The reason is that this is a linearized model around the equilibrium point. Around the equilibrium point, this does not cause large deviation. While when the pitch angle becomes large, an increase in elevation rate is clearly accompanied by an increase in travel rate. As this is impossible to model with a linear model, additional constraints are imposed in later task.

6.2 Discretization

Use forward Euler method, the same as euaqtion (4.7), the discretized state-space equation is

$$\begin{aligned} \mathbf{x}_{k+1} = & \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & -\Delta t K_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & -\Delta t K_1 K_{pp} & 1 - \Delta t K_1 K_{pd} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & -\Delta t K_e K_{ep} & 1 - \Delta t K_3 K_{ed} \end{bmatrix} \mathbf{x}_k \\ & + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t K_1 K_{pp} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \Delta t K_3 K_{ep} \end{bmatrix} \mathbf{u}_k \end{aligned} \tag{6.2}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 1 & -0.1415 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.25 & 0 & 0 \\ 0 & 0 & -0.81 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 0 & -0.16 & 0.6 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.81 & 0 \\ 0 & 0 \\ 0 & 0.16 \end{bmatrix} \tag{6.3}$$

6.3 Optimal trajectory with nonlinear constraints

The criteria to be minimized is now

$$\begin{aligned} \phi &= \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + q_1 p_{ci}^2 + q_2 e_{ci}^2 \\ &= \sum_{i=1}^{N} (\mathbf{x_i}^T \mathbf{Q} \mathbf{x_i} + \mathbf{u_i}^T \mathbf{R} \mathbf{u_i}) \end{aligned} \tag{6.4}$$

For every point in time a constraint is implemented on the form

$$c(x_k) = \alpha \exp(-\beta(\lambda_k - \lambda_t)^2) - e_k \leq 0 \tag{6.5}$$

where $\alpha = 0.2$, $\beta = 20$ and $\lambda_t = \frac{2\pi}{3}$. The optimization problem can be ststed as

$$\underset{\mathbf{z}}{min} \ \mathbf{z}^T \mathbf{G} \mathbf{z} \tag{6.6}$$

subject to

$$\begin{aligned} &\mathbf{A}_{eq} \mathbf{z} = \mathbf{B}_{eq}, \\ &c(x_k) \leq 0, \ k \in \{1, ..., N\} \\ &p^{low} \leq p_k, \ k \in \{1, ..., N\} \\ &p_k \leq p^{high}, \ k \in \{1, ..., N\} \end{aligned} \tag{6.7}$$

6.4 Result and Discussion

6.4.1 Open-loop result

The calculation result and simulation result without feedback is compared below. It shows that the helicopter can follow the optimal path in the beginning but cannot reach the reference point eventually.

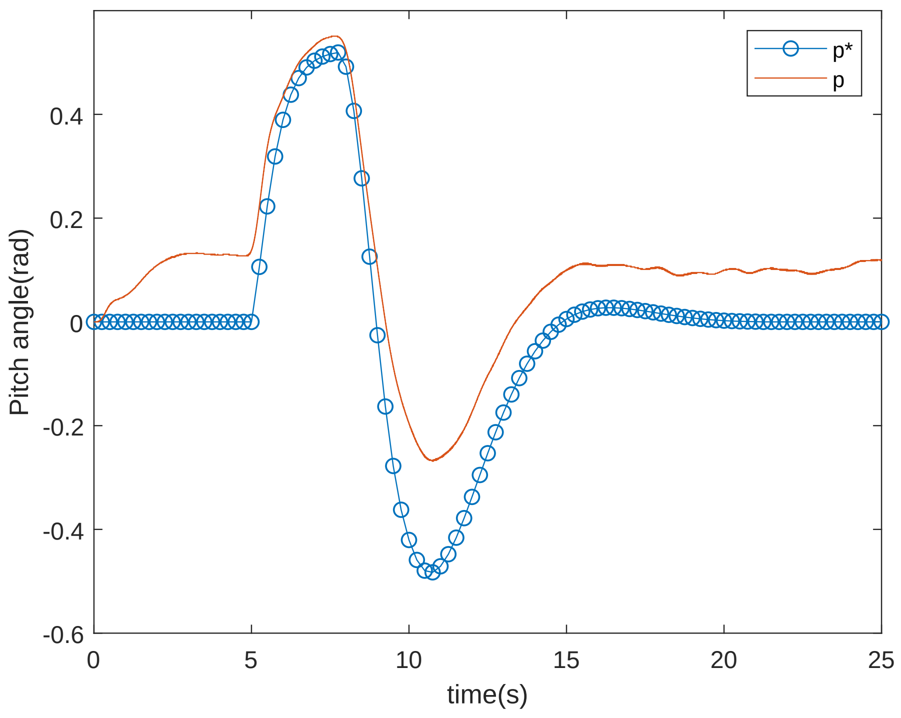

Figure 10: Measured and optimal pitch angle without feedback

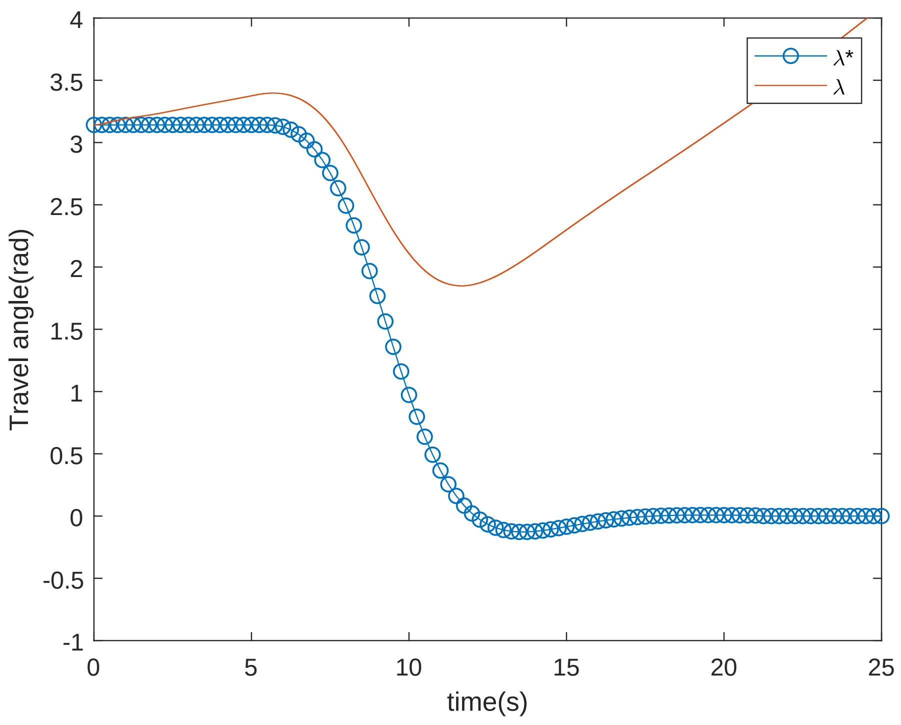

Figure 11: Measured and optimal travel angle without feedback

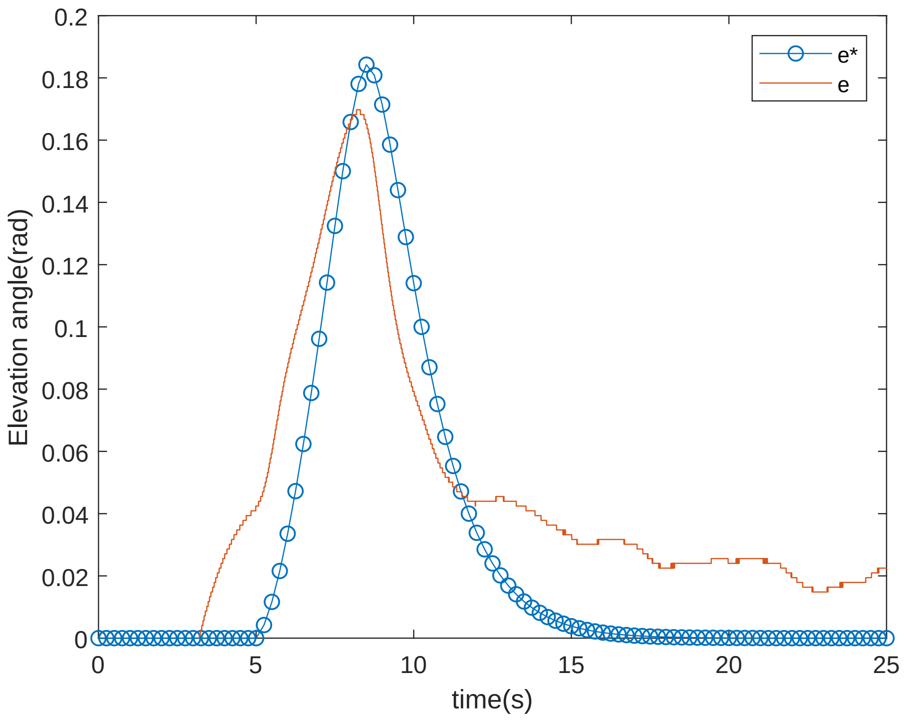

Figure 12: Measured and optimal elevation angle without feedback

6.4.2 Close-loop result

The weighting matrix is tuned as below

$$\mathbf{Q} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \tag{6.8}$$

The resulting K-matrix becomes

$$\mathbf{K} = \begin{bmatrix} -2.272 & -5.575 & 2.317 & 0.535 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.982 & 1.261 \end{bmatrix} \tag{6.9}$$

With feedback, the simulation result is much better than the open-loop result. It can properly follow the optimal path calculated by MATLAB, while the pitch angle still deviates from the optimal trajectory as it does in 5.2. Also, the helicopter can reach the desired point in the end but still shows a small drift in the beginning of switch on and after reaching that point. This is reasonable since the state-space model is not so perfect that can exactly model the helicopter and make it follow the optimal path. This is again due to the fact that the pitch and elevation are decoupled in the model used.

Also, what needs to mention is that if we choose $N = 40$, it shows a sharp variety in travel at $t = 15s$. The reason is that within limited time horizon, the path cannot be optimized properly. If $N = 65$ is chosen, the result becomes much better and smoother than before.

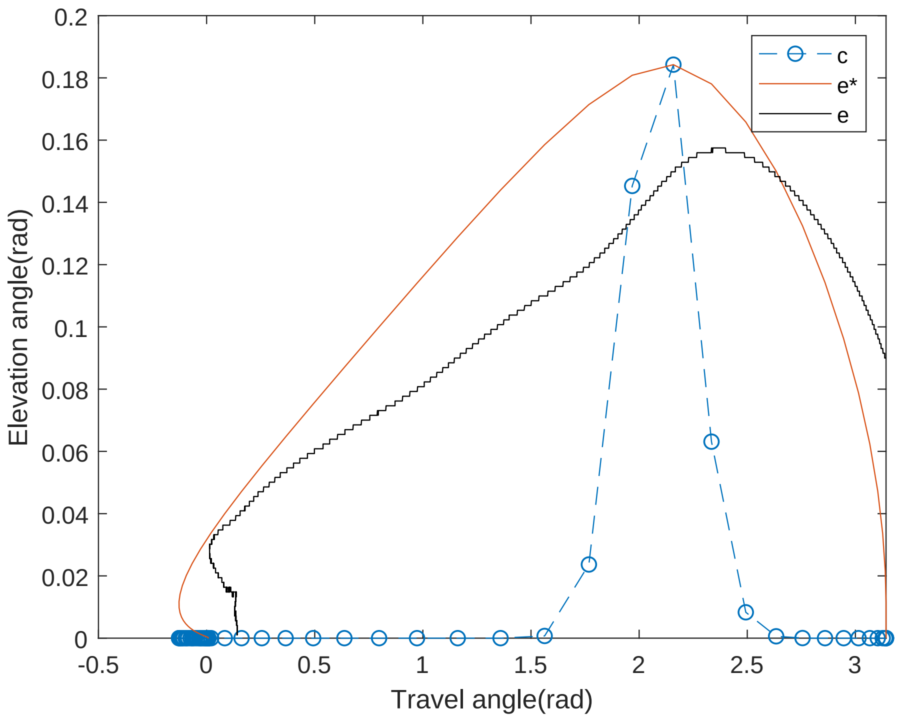

Figure 13: LQR constraint

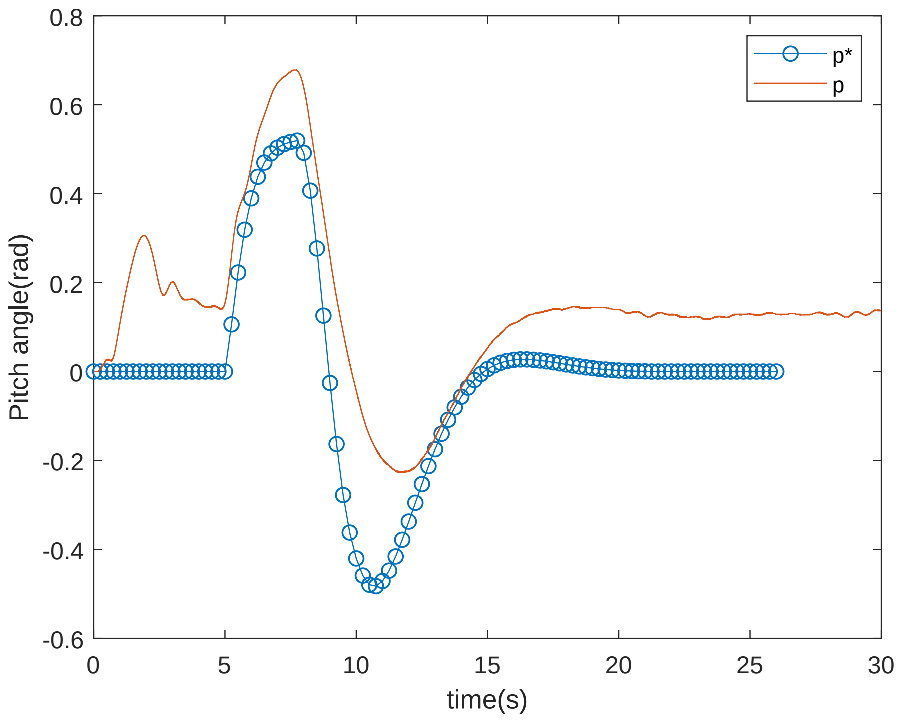

Figure 14: Measured and optimal pitch angle with feedback

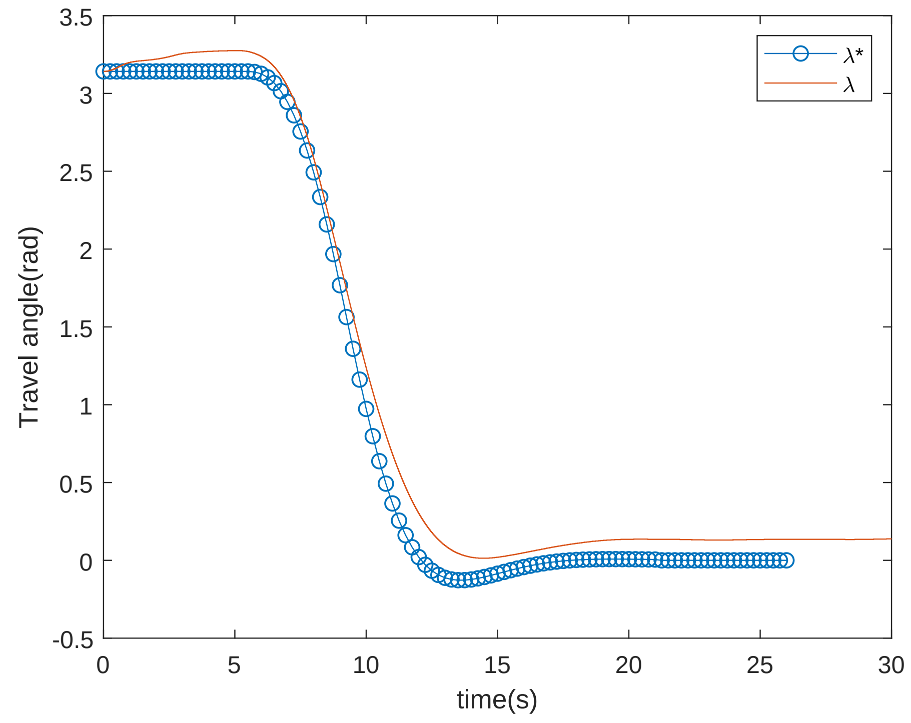

Figure 15: Measured and optimal travel angle with feedback

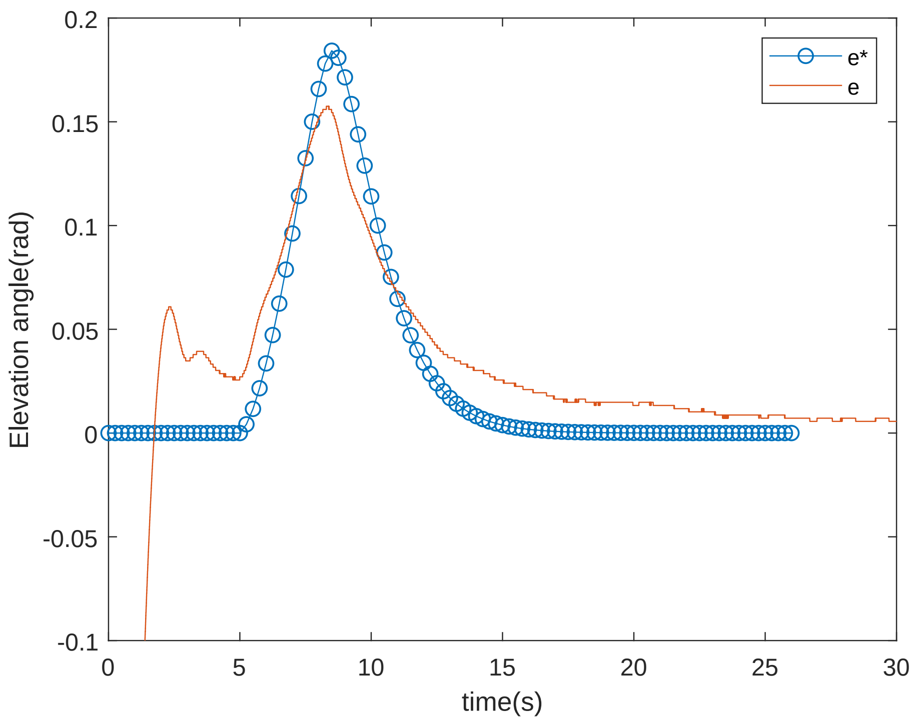

Figure 16: Measured and optimal elevation angle with feedback

6.4.3 Alternative Constraints

In this part, a more strict constraint is implemented. Since in the real helicopter model, the motion is coupled and our linearised model cannot model the real helicopter motion properly. The solution is to keep the system within the linear region: the small motion and motion rate region.

$$\begin{aligned} |p| &\leq \frac{30\pi}{180},\ k \in \{1, ..., N\} \\ |\dot{\lambda}| &\leq 0.6,\ k \in \{1, ..., N\} \\ |\dot{e}| &\leq 0.075,\ k \in \{1, ..., N\} \end{aligned} \tag{6.10}$$

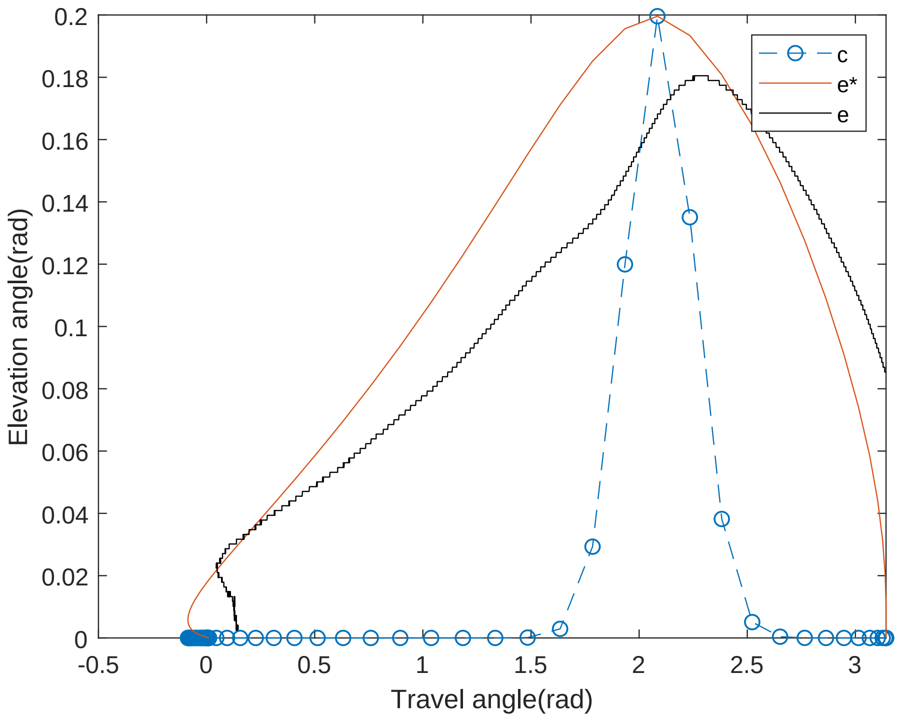

Figure 17: LQR constraint

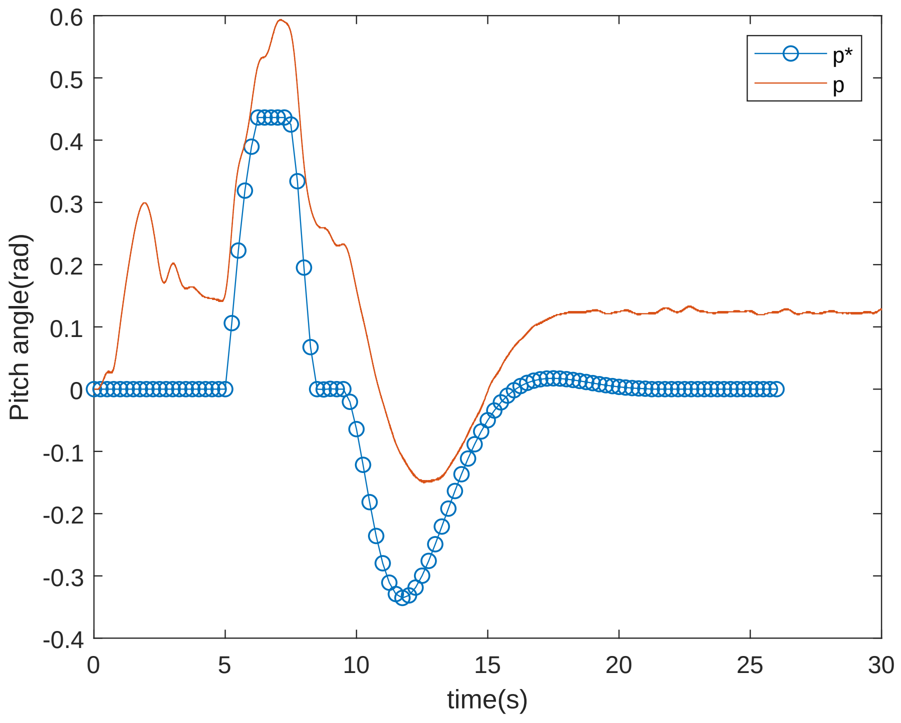

Figure 18: Measured and optimal pitch angle with feedback

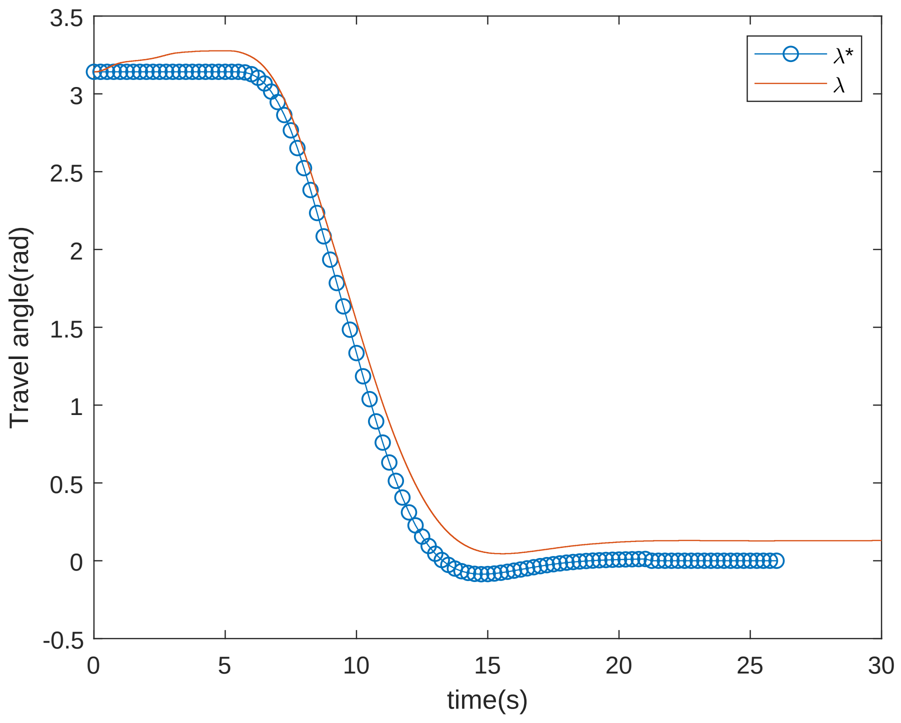

Figure 19: Measured and optimal travel angle with feedback

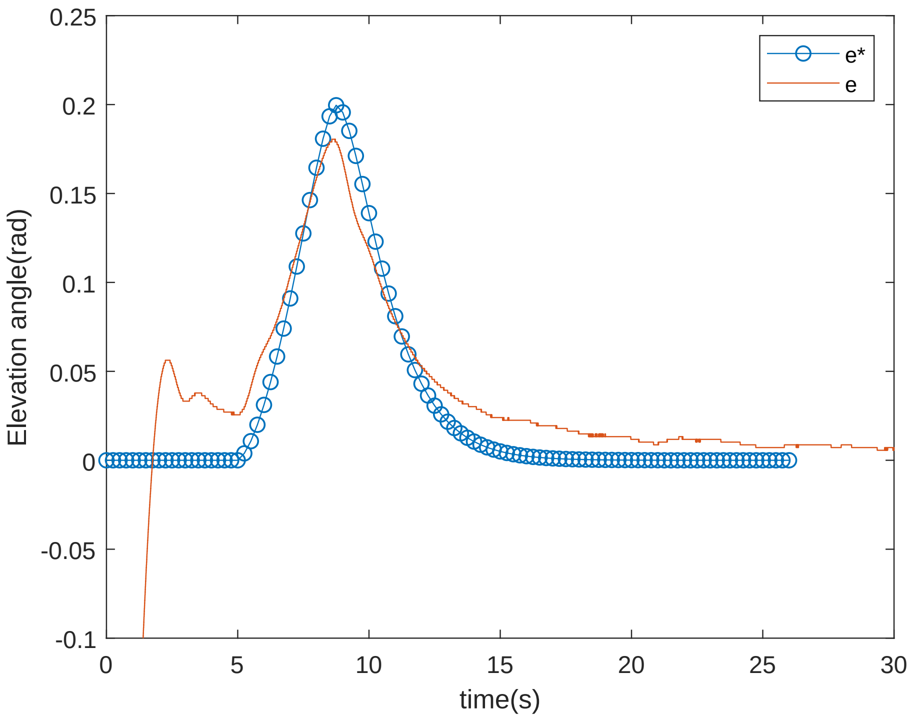

Figure 20: Measured and optimal elevation angle with feedback

A comparison between the open loop, closed loop and alternative closed loop responses is shown above. It is clear that the travel and elevation angle follows batter to the optimized one. This proves that by restricting the motion in smaller velocity, the linearised model can better simulate the real helicopter model.

It should be noticed that both with or without additional constraints, the nonlinear constraints are violated, while constraint violation is reduced when additional constraints are imposed. The reason of violation might be uncoupling effect in the state space model. As the pitch angle increase, the real model would have increase of elevation. Also, with state feedback, the trajectory is corrected with respect to the optimal path in time, while the bound on elevation is defined with respect to travel. This means that a deviation in travel will shift the optimal elevation path with respect to travel.

In order to minimized these effect, additional constraints are imposed on elevation rate and travel rate. With these extra constraints, we have a smoother optimal elevation trajectory, and less aggressive system, and thus less violation.

7 Discussion and Conclusion

The helicopter control is designed with and without feedback by optimizing the path of $[\lambda, p]$ and the path of $[\lambda, p, e]$ via minimizing the cost function under constraints. In the first, the controller is designed without feedback by taking pitch and travel motion into account, which generates large deviation from the calculated optimal trajectory. The helicopter cannot reach the desired position in the end. When the feedback is added to this system, a tuned LQ controller enables the system to perform a lot better than before. A discussion of MPC(model predictive control) and LQ controller is also done based on both the cost and optimization performance, the later of which is recommended in this project since little disturbance exists and the helicopter requires fast dynamics indeed.

Then, the elevation motion with a nonlinear constraint is also considered into the controller design. Without feedback, the system cannot follow the optimized path again. But with feedback, the helicopter performs little better than without elevation control. Therefore, from the tests above, it is reasonable to say that the implementation of feedback control is vital to cancel the deviation and realize path optimization and better performed optimization relays on taking more states ino consideration. Finally, a more strict constraint is tested to add in the optimization model, which gives a slightly more accurate response.

8 MATLAB Script

Listing 1: Optimal Control of Pitch/Travel without Feedback

```
%% Initial matrix
A_c = [0 1 0 0; 0 0 -K_2 0; 0 0 0 1; 0 0 -K_1*K_pp -K_1*K_pd];
B_c = [0;0;0;K_1*K_pp];
delta_t = 0.25;
A = eye(4) + delta_t*A_c;
B = delta_t*B_c;

%% Number of states and inputs
nx = size(A,2);
nu = size(B,2);

%% Time horizon and initialization
N = 100;
z = zeros(N*(nx+nu),1);
x0_1 = pi;
x0_2 = 0;
x0_3 = 0;
x0_4 = 0;
x0 = [x0_1 x0_2 x0_3 x0_4]';
xf = [0 0 0 0]';

%% Upper and lower bounds
u_l = -30*pi/180;
u_u = 30*pi/180;
x_l = [-inf -inf -30*pi/180 -inf]';
x_u = [inf inf 30*pi/180 inf]';

LB_x = repmat( x_l, N, 1);
UB_x = repmat( x_u, N, 1);
LB_u = repmat( u_l, N, 1);
UB_u = repmat( u_u, N, 1);
LB = [LB_x; LB_u];
UB = [UB_x; UB_u];


%% Solve QP
Q = zeros(nx);
Q(1,1) = 1;
q = 1;
R = q;
G = blkdiag(kron(eye(N),Q),kron(eye(N),R));

A_eq = [eye(N*nx)+kron(diag(ones(N-1,1),-1),-A), kron(eye(N), -B)];
b_eq = [A*x0; zeros(nx*(N-1),1)];

[z,fval,exitflag,output,lambda] = quadprog(G, [], [], [], A_eq, b_eq, LB, UB);

%% Extract states and control input
x = [reshape(z(1:N*nx),[nx,N])];
u = [reshape(z(N*nx+1:end),[nu,N])];

%% Prep input sequence
padding_time = 5;
padding_input = [zeros(1,floor(padding_time/delta_t)),u,zeros(1,floor(padding_time/delta_t))
    ]';
time = [(0:length(padding_input)-1)*delta_t]';
heli_input = [time padding_input];

travel_opt = [x0_1*ones(1,floor(padding_time/delta_t)), x(1,:), zeros(1,floor(padding_time/
    delta_t))];
travel_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(2,:), zeros(1,floor(padding_time/
    delta_t))];
pitch_opt = [zeros(1,floor(padding_time/delta_t)), x(3,:), zeros(1,floor(padding_time/delta_t)
    )];
pitch_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(4,:), zeros(1,floor(padding_time/
    delta_t))];
```

Listing 2: Optimal Control of Pitch/Travel with Feedback (LQ)

```
% Extract states and control input
x = [reshape(z(1:N*nx),[nx,N])];
u = [reshape(z(N*nx+1:end),[nu,N])];

% Prep input sequence
padding_time = 5;
padding_input = [zeros(1,floor(padding_time/delta_t)),u,zeros(1,floor(padding_time/delta_t))
    ]';
time = [(0:length(padding_input)-1)*delta_t]';
heli_input = [time padding_input];

travel_opt = [x0_1*ones(1,floor(padding_time/delta_t)), x(1,:), zeros(1,floor(padding_time/
    delta_t))];
travel_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(2,:), zeros(1,floor(padding_time/
    delta_t))];
pitch_opt = [zeros(1,floor(padding_time/delta_t)), x(3,:), zeros(1,floor(padding_time/delta_t)
    )];
pitch_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(4,:), zeros(1,floor(padding_time/
    delta_t))];

% LQR control
Q_c = diag([8 1 0 0]);
% R_c = 0.01
% R_c = 0.05
R_c = 1;

K = dlqr(A,B,Q_c,R_c);

state_opt = [travel_opt; travel_rate_opt; pitch_opt; pitch_rate_opt];
heli_state = [time state_opt'];
```

Listing 3: Optimal Control of Pitch/Travel and Elevation without Feedback

```
%% Initial matrix
A_c = [0 1 0 0 0 0;
       0 0 -K_2 0 0 0;
       0 0 0 1 0 0;
       0 0 -K_1* K_pp -K_1* K_pd 0 0;
       0 0 0 0 0 1;
       0 0 0 0 -K_3*K_ep -K_3*K_ed];
B_c = [0 0; 0 0; 0 0; K_1*K_pp 0; 0 0; 0 K_3*K_ep];
delta_t = 0.25;
A = eye(6) + delta_t*A_c;
B = delta_t*B_c;

%% Number of states and inputs
nx = size(A,2);
nu = size(B,2);

%% Time horizon and initialization
N = 65;
z0 = zeros(N*(nx+nu),1);
z0(1) = pi;
x0_1 = pi;
x0_2 = 0;
x0_3 = 0;
x0_4 = 0;
x0_5 = 0;
x0_6 = 0;
x0 = [x0_1 x0_2 x0_3 x0_4 x0_5 x0_6]';
xf = [0 0 0 0 0 0]';

%% Upper and lower bounds
u_l = [-30*pi/180 -inf]';
u_u = [30*pi/180 inf]';
x_l = [-inf -inf -30*pi/180 -inf -inf -inf]';
x_u = [inf inf 30*pi/180 inf inf inf]';

LB_x = repmat( x_l, N, 1);
UB_x = repmat( x_u, N, 1);
LB_u = repmat( u_l, N, 1);
UB_u = repmat( u_u, N, 1);
```

```
LB = [LB_x; LB_u];
UB = [UB_x; UB_u];


%% Solve QP
Q = zeros(nx);
Q(1,1) = 1;
R = diag([1,1]);
G = blkdiag(kron(eye(N),Q),kron(eye(N),R));

A_eq = [eye(N*nx)+kron(diag(ones(N-1,1),-1),-A), kron(eye(N), -B)];
b_eq = [A*x0; zeros(nx*(N-1),1)];

f = @(z) z'*G*z;


options=optimoptions('fmincon','Display','iter','Algorithm','sqp');
[z,fval] = fmincon(f,z0,[],[],A_eq,b_eq,LB,UB,@nonlcon,options);


%% Extract states and control input
x = [reshape(z(1:N*nx),[nx,N])];
u = [reshape(z(N*nx+1:end),[nu,N])];

%% Prep input sequence
padding_time = 5;
padding_input = [zeros(2,floor(padding_time/delta_t)),u,zeros(2,floor(padding_time/delta_t))
    ]';
time = [(0:length(padding_input)-1)*delta_t]';
heli_input = [time padding_input];

travel_opt = [x0_1*ones(1,floor(padding_time/delta_t)), x(1,:), zeros(1,floor(padding_time/
    delta_t))];
travel_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(2,:), zeros(1,floor(padding_time/
    delta_t))];
pitch_opt = [zeros(1,floor(padding_time/delta_t)), x(3,:), zeros(1,floor(padding_time/delta_t)
    )];
pitch_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(4,:), zeros(1,floor(padding_time/
    delta_t))];
ele_opt = [zeros(1,floor(padding_time/delta_t)), x(5,:), zeros(1,floor(padding_time/delta_t))
    ];
ele_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(6,:), zeros(1,floor(padding_time/
    delta_t))];

state_opt = [travel_opt; travel_rate_opt; pitch_opt; pitch_rate_opt; ele_opt; ele_rate_opt];
heli_state = [time state_opt'];
```

Listing 4: Optimal Control of Pitch/Travel and Elevation with Feedback

```
%% Extract states and control input
x = [reshape(z(1:N*nx),[nx,N])];
u = [reshape(z(N*nx+1:end),[nu,N])];

%% Prep input sequence
padding_time = 5;
padding_input = [zeros(2,floor(padding_time/delta_t)),u,zeros(2,floor(padding_time/delta_t))
    ]';
time = [(0:length(padding_input)-1)*delta_t]';
heli_input = [time padding_input];

travel_opt = [x0_1*ones(1,floor(padding_time/delta_t)), x(1,:), zeros(1,floor(padding_time/
    delta_t))];
travel_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(2,:), zeros(1,floor(padding_time/
    delta_t))];
pitch_opt = [zeros(1,floor(padding_time/delta_t)), x(3,:), zeros(1,floor(padding_time/delta_t)
    )];
pitch_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(4,:), zeros(1,floor(padding_time/
    delta_t))];
ele_opt = [zeros(1,floor(padding_time/delta_t)), x(5,:), zeros(1,floor(padding_time/delta_t))
    ];
ele_rate_opt = [zeros(1,floor(padding_time/delta_t)), x(6,:), zeros(1,floor(padding_time/
    delta_t))];

```

```
state_opt = [travel_opt; travel_rate_opt; pitch_opt; pitch_rate_opt; ele_opt; ele_rate_opt];
heli_state = [time state_opt'];

% LQR
Q_c = diag([4,1,5,5,5,0]);
R_c = diag([0.5,0.5]);
K = dlqr(A,B,Q_c,R_c);
```

Listing 5: MTLAB function `nonlcon`

```
function [c,c_eq] = nonlcon(z)
%Nonlinear constraints

alpha = 0.2;
beta = 20;
lambda_t = 2*pi/3;
n_x = 6;
N = 65;

lambda_k = z(1:n_x:N*n_x );
e_k = z(5:n_x:N*n_x );

c = zeros(N,1);

for i = 1:N
c(i) = alpha*exp(-beta *(lambda_k(i)-lambda_t)^2)-e_k(i);
end

c_eq= [];

end
```

Listing 6: Optimal Control of Pitch/Travel and Elevation with alternative constraints

```
%% Upper and lower bounds
u_l = [-30*pi/180 -inf]';
u_u = [30*pi/180 inf]';
x_l = [-inf -0.6 -30*pi/180 -inf -inf -0.075]';
x_u = [inf 0.6 30*pi/180 inf inf 0.075]';
```

9 Simulink Diagrams

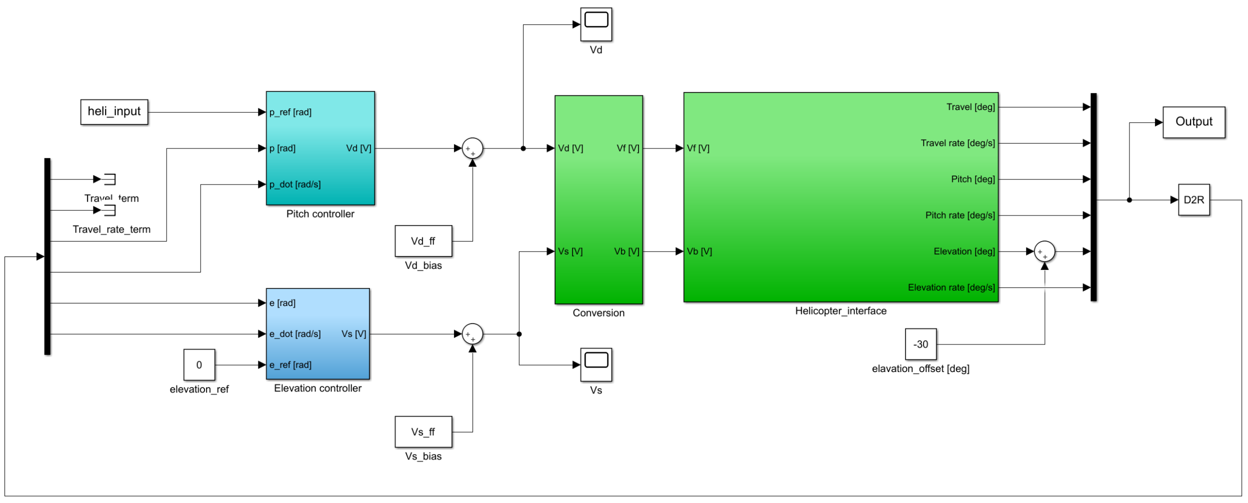

Figure 21: Simulink diagram in section 4

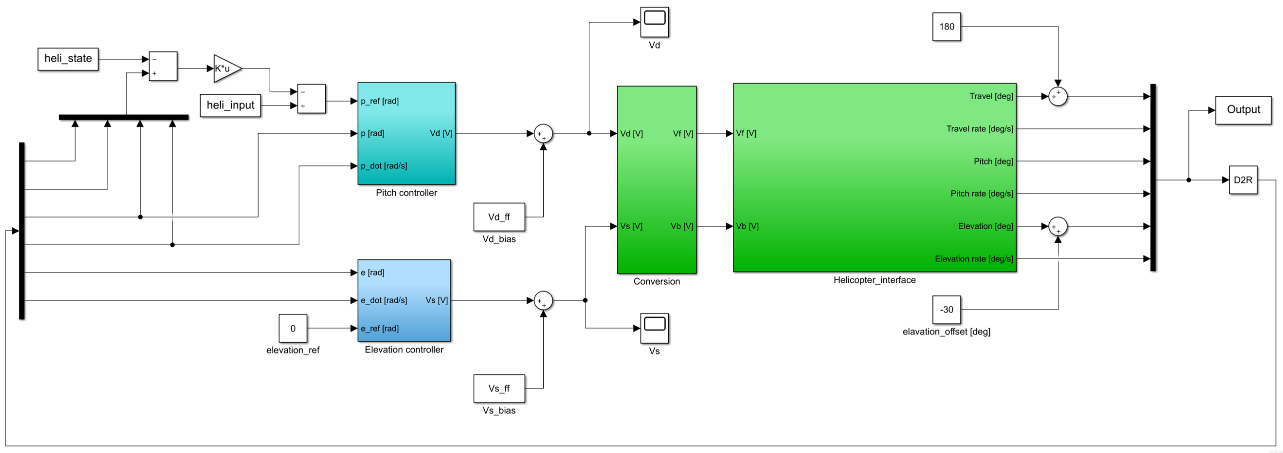

Figure 22: Simulink diagram in section 5

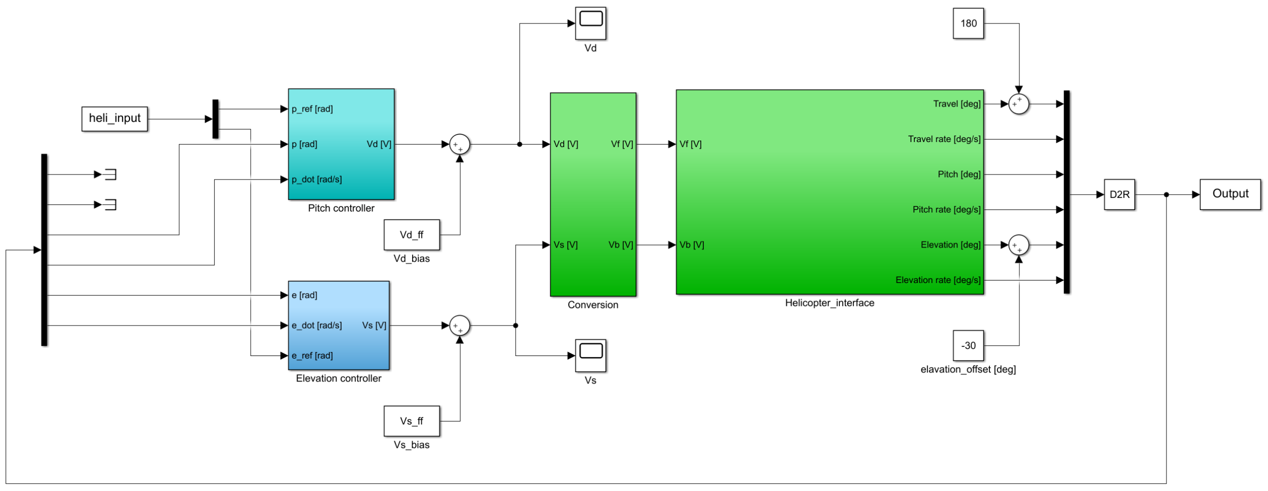

Figure 23: Open-loop Simulink diagram in section 6

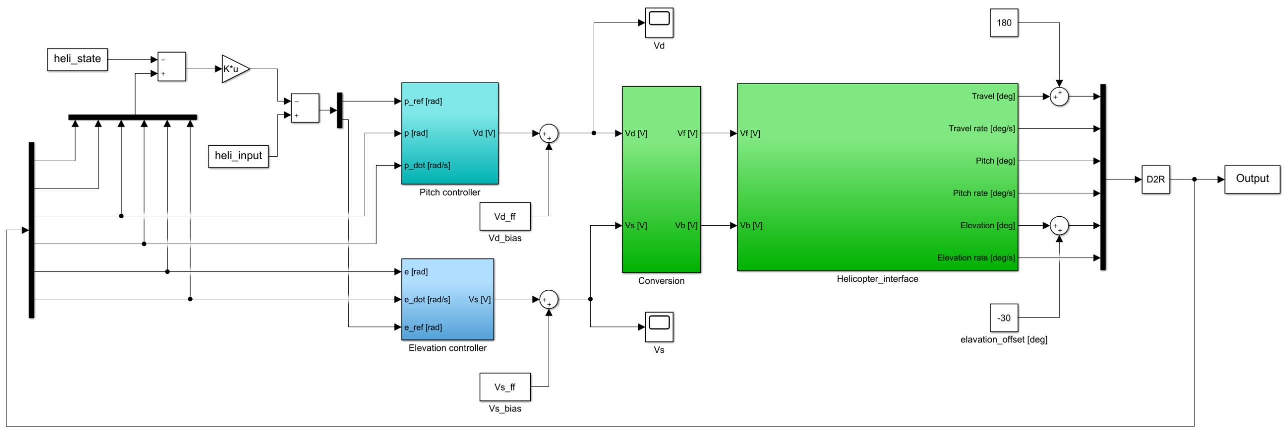

Figure 24: Close-loop Simulink diagram in section 6

References

[1] Chi-Tsong Chen, *Linear System Theory and Design.* Oxford University Press, international fourth edition, 2013.

[2] Nocedal Wright, *Numerical Optimization.* 2nd Edition, 2006.