TTK4135 Optimization and Control
Lab Report

716120, 723987
Group 1

April 22, 2015

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Abstract

This document outlines a few important aspects of the lab report. It contains some advice on both content and layout. The Latex source for this document is also published, and you can use it as a template of sorts for your own report.

When you write your own report, this section (the abstract) should contain a *very* short summary of what the lab is about and what you have done.

Contents

Abstract **2**

Contents **3**

1 Introduction **4**

2 Problem Description **5**

3 Repetition/Introduction to Simulink/QuaRC **8**
- 3.1 PID-(re)tuning . . . 8
- 3.2 Results and discussion . . . 8

4 Optimal Control of Pitch/Travel without Feedback **9**
- 4.1 State space model . . . 9
- 4.2 Discretization . . . 9
- 4.3 Optimal trajectory . . . 10
- 4.4 Results and discussion . . . 11

5 Optimal Control of Pitch/Travel with Feedback (LQ) **12**
- 5.1 Discrete LQR . . . 12
- 5.2 Results and discussion . . . 12
  - 5.2.1 MPC discussion . . . 12

6 Optimal Control of Pitch/Travel and Elevation with and without Feedback **14**
- 6.1 State space model . . . 14
- 6.2 Discretization . . . 14
- 6.3 Optimization problem with nonlinear constraints . . . 14
- 6.4 Optional: Additonal constraints . . . 15
- 6.5 Results and discussion . . . 15

7 Discussion **19**

8 Conclusion **20**

A MATLAB Code **21**
- A.1 System identification . . . 21

B Simulink Diagrams **23**
- B.1 Open loop optimization . . . 23

1 Introduction

Your introduction should contain an overview of the work you were assigned, as well as a few sentences putting the work into a larger perspective. You should also give a quick description of how the report is organized (as is done below).

2 Problem Description

The lab setup consists of a movable arm equipped with two rotors. The movable arm is hinged to a fixed point, allowing for both lateral and longitudinal motion. The arm is also fitted with a counterweight which effectively slows the dynamics down considerably, as well as lower the amount of rotor thrust needed. The two rotors are fixed to a pitch head assembly hinged to the movable arm. This allows the rotor thrust direction to be indirectly controlled by the differential thrust applied.

From first principles analysis we can derive simple differential equations to describe the system dynamics about the equilibrium:

$$\ddot{p} = K_1 V_d, \qquad K_1 = \frac{K_f l_h}{J_p}, \tag{1a}$$

$$\ddot{\lambda} = -K_2 p, \qquad K_3 = \frac{K_f l_a}{J_e} \tag{1b}$$

$$\ddot{e} = K_3 V_s - \frac{T_g}{J_e}, \qquad K_2 = \frac{K_p l_a}{J_t} \tag{1c}$$

Note simplifications and limitations:

- The time derivative of travel rate is a linear function of pitch only. This small angle approximation does not really hold, as the intended operating range of pitch is as much as 40 degrees.
- By simple inspection of the lab setup it is clear that the pitch head assembly is hinged slightly above its center of mass. The resulting restoring force, as well as the hinge joint dampening, is not directly included in this model.

To stabilize the plant, adding the pitch PD controller

$$V_d = K_{pp}(p_c - p) - K_{pd}\dot{p}, \quad K_{pp}, K_{pd} > 0$$

and the elevation PID controller

$$V_s = K_{ei} \int (e_c - e)\ dt + K_{ep}(e_c - e) - K_{ed}\dot{e}, \quad K_{ei}, K_{ep}, K_{ed} > 0$$

yields the model equations

$$\ddot{e} + K_3 K_{ed}\dot{e} + K_3 K_{ep} e = K_3 K_{ep} e_c \tag{2a}$$

$$\ddot{p} + K_1 K_{pd}\dot{p} + K_1 K_{pp} p = K_1 K_{pp} p_c \tag{2b}$$

$$\ddot{\lambda} = -K_2 p \tag{2c}$$

where it is assumed the elevation integral term counteracts the constant disturbance $-\frac{T_g}{J_e}$ and cancel out.

However, in order to achieve a more accurate model we can utilize statistical procedures to best identify the parameters of the first and second order systems. This eliminates errors present in

Table 1: Parameters and values.

Symbol	Parameter	Value	Unit
l_a	Distance from elevation axis to helicopter body	0.63	m
l_h	Distance from pitch axis to motor	0.18	m
K_f	Force constant motor	0.25	N/V
J_e	Moment of inertia for elevation	0.83	kg m^2
J_t	Moment of inertia for travel	0.83	kg m^2
J_p	Moment of inertia for pitch	0.034	kg m^2
m_h	Mass of helicopter	1.05	kg
m_w	Balance weight	1.87	kg
m_g	Effective mass of the helicopter	0.05	kg
K_p	Force to lift the helicopter from the ground	0.49	N

Table 2: Variables

Symbol	variable
p	Pitch
p_c	Pitch setpoint
λ	Travel
λ_c	Travel rate setpoint
e	Elevation
e_c	Elevation setpoint
V_f	Voltage input, front motor
V_b	Voltage input, back motor
V_d	Voltage difference, $V_f - V_b$
V_s	Voltage sum, $V_f + V_b$
$K_{pp}, K_{pd}, K_{ep}, K_{ei}, K_{ed}$	Controller gains
T_g	Torque exerted by gravity

the measurement of the lab setup, and gives us the best model with the given number of states. This system gray-box system identification also allows us to verify that the proposed number of states, derived by first principles, yields a model whose performance matches that of the actual system.

3 Repetition/Introduction to Simulink/QuaRC

3.1 PID-(re)tuning

The pre-tuned PID showed unsatisfactory performance and was re-tuned to better serve as the stable plant for the rest of the assignement.

If we want a table comparing the gains...

Table 3: Controller gains comparison

Gain	Original	New
K_{pp}	93.2	14.0
K_{pd}	13.2	2.5
K_{ei}	2.3	2.3
K_{ep}	7.0	15.0
K_{ed}	10.0	13.0

3.2 Results and discussion

Figure 1 shows the pitch and elevation response to a step input, as well as the resulting travel rate to a step pitch input. Would be nice to have a plot of the step responses with the original gains, to compare...

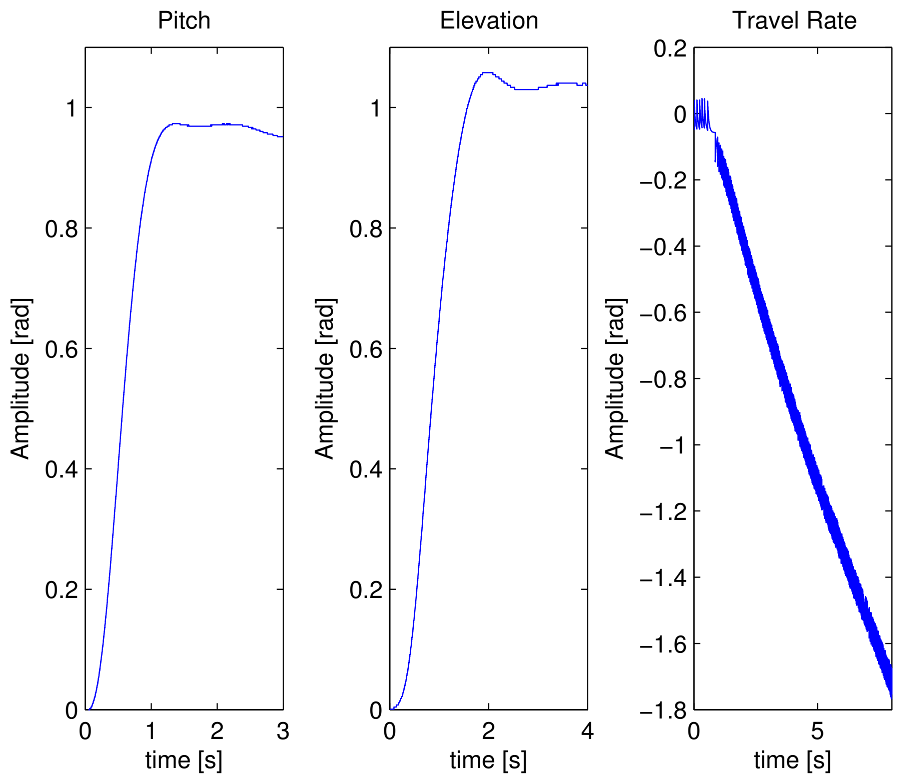

Figure 1: Step response.

4 Optimal Control of Pitch/Travel without Feedback

4.1 State space model

From the model equations in (2) we get the continuous state space equation

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -K_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_1 K_{pp} & -K_1 K_{pd} \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_1 K_{pp} \end{bmatrix} p_c$$

or, alternatively

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.0663 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -5.3095 & -0.9481 \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 5.3095 \end{bmatrix} p_c$$

However, in an effort to achieve a more accurate model, alternative state space equations are developed from estimated transfer functions based on measured step response output.

Discuss: 2 pole vs 3 pole, include transfer functions...Not exactly sure exactly what decided on, and how the model got these parameters...

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.03 & -0.39 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -7.13 & -3.6 \end{bmatrix}}_{A_c} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 6.74 \end{bmatrix}}_{B_c} p_c \tag{3}$$

asdfDiscuss differences with textbook model

4.2 Discretization

Let $x = \begin{bmatrix} \lambda & \dot{\lambda} & p & \dot{p} \end{bmatrix}^\top$, $u = p_c$. Using forward Euler with a time step $\Delta t = 0.25$ we are able to obtain an approximate discretization of (3):

$$\begin{align} x_{k+1} &= x_k + \Delta t \dot{x}_k \tag{4a} \\ &= x_k + \Delta t (A_c x_k + B_c u_k) \tag{4b} \\ &= (I + \Delta t A_c) x_k + (\Delta t B_c) u_k \tag{4c} \\ &= A x_k + B u_k \tag{4d} \end{align}$$

where $x_k = x(k\Delta t) \in \mathbb{R}^{n_x}$, $u_k = u(k\Delta t) \in \mathbb{R}^{n_u}$, and

$$A = \begin{bmatrix} 1 & 0.25 & 0 & 0 \\ 0 & 0.9925 & -0.0975 & 0 \\ 0 & 0 & 1 & 0.25 \\ 0 & 0 & -1.7825 & 0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.685 \end{bmatrix}. \tag{5}$$

4.3 Optimal trajectory

We calculate the trajectory from $x_0 = \begin{bmatrix}\lambda_0 & 0 & 0 & 0\end{bmatrix}^\top$ to $x_f = \begin{bmatrix}\lambda_f & 0 & 0 & 0\end{bmatrix}^\top$ minimizing the objective function

$$\phi = \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + r p_{c_i}^2, \quad r \geq 0, \tag{6}$$

where we let $\lambda_0 = 0$, $\lambda_f = \pi$.

The parameter r weights the relative importance of low input expendature, in this case set-point for the pitch angle, versus a rapid convergence of the travel trajectory to λ_f. Equivalently we can define (6) in terms of the full state and input variables.

$$\phi = \frac{1}{2} \sum_{i=0}^{N-1} (x_{i+1} - x_f)^\top Q (x_{i+1} - x_f) + u_i^\top R u_i, \tag{7}$$

where

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = r.$$

The system dynamics (4) subjects (7) to the linear equality constraints

$$\underbrace{\left[\begin{array}{cccc|cccc} I & & & & -B & & & \\ -A & \ddots & & & & \ddots & & \\ & \ddots & \ddots & & & & \ddots & \\ & & -A & I & & & & -B \end{array}\right]}_{A_{eq} \in \mathbb{R}^{N n_x \times N(n_x+n_u)}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{z \in \mathbb{R}^{N(n_x+n_u)\times 1}} = \underbrace{\begin{bmatrix} -A x_f \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B_{eq} \in \mathbb{R}^{N n_x \times 1}}. \tag{8}$$

To express (7) in terms of the optimization variable z we define the matrix $G \in \mathbb{R}^{N(n_x+n_u)\times N(n_x+n_u)}$:

$$G = \begin{bmatrix} Q & & & & & \\ & \ddots & & & & \\ & & Q & & & \\ & & & R & & \\ & & & & \ddots & \\ & & & & & R \end{bmatrix}.$$

With lower and upper bounds imposed on the pitch state and controller set-point, the QP-problem can then be stated:

$$\min_{z} \quad \frac{1}{2} z^\top G z \tag{9a}$$

subject to

$$A_{eq} z = B_{eq}, \tag{9b}$$

$$p^{\text{low}} \leq p_k \leq p^{\text{high}}, \quad k \in \{1, \dots, N\}. \tag{9c}$$

4.4 Results and discussion

(9) is solved using MATLAB's `quadprog`. $r = 0.1$ is chosen to achieve a relatively rapid convergence rate in with the effect of maximizing the pitch between the lower and higher bounds. The optimal input sequence u^* is applied to the plant in an open loop with results show in figure 2, with the measured trajectory compared to the calculated optimal trajectory x^*. For some clever reason or another, the pitch bounds was set to $\frac{45\pi}{180}$.

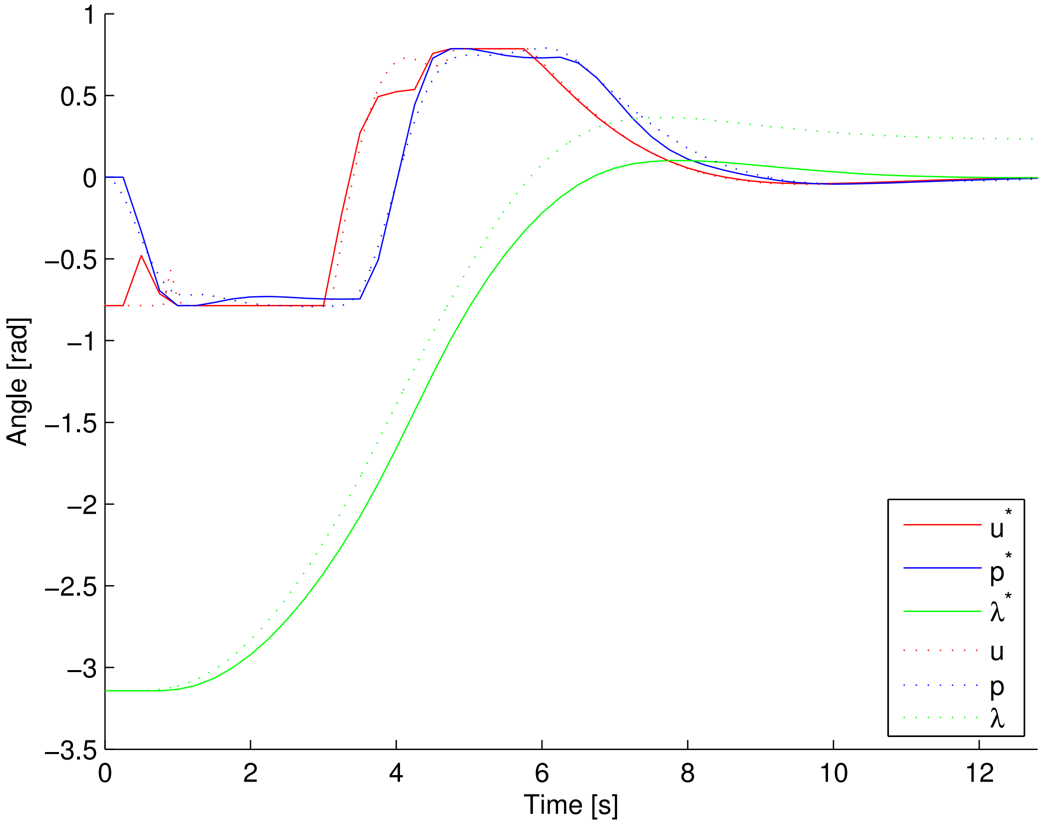

Figure 2: Optimal vs. measured trajectory and input sequence.

Opt. input vs measured input discrepency due to interpolation stuff...?

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

5.1 Discrete LQR

To eliminate the discrepency between the optimal and measured trajectory observed in figure 2, we can update the optimal trajectory for every time step with a state feedback term weighted by a suitable gain matrix K:

$$u_k = u_k^* - K^\top (x_k - x_k^*),$$

or, alternatively

$$\Delta u_k = -K^\top \Delta x_k, \tag{10}$$

where

$$\begin{aligned} \Delta x_k &= x_k - x_k^*, \\ \Delta u_k &= u_k - u_k^*. \end{aligned}$$

It can be shown (some clever reference here...) that the controller (10) is the optimal solution minimizing the quadratic objective function

$$J = \sum_{i=0}^{\infty} \Delta x_{i+1}^\top \tilde{Q} \Delta x_{i+1} + \Delta u_i^\top \tilde{R} \Delta u_i,$$

subject to the system dynamics (4), where

$$K = (R + B^\top P B)^{-1} B^\top P A, \tag{11}$$

and P is the unique positive definite solution to the discrete time algebraic Riccati equation.(11) is used as the state feedback gain, and the resulting Linear Quadratic controller is implemented, with weigthing matrices Q and R chosen to penalize deviations in states and input for a satisfactory results.

5.2 Results and discussion

Maybe discuss (L)QR (pun intended), tuning...

$$\tilde{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad \tilde{R} = 0.2.$$

Using MATLAB's `dlqr` we obtain the LQ state feedback gain, and the controller is applied with results shown in figure 3.

5.2.1 MPC discussion

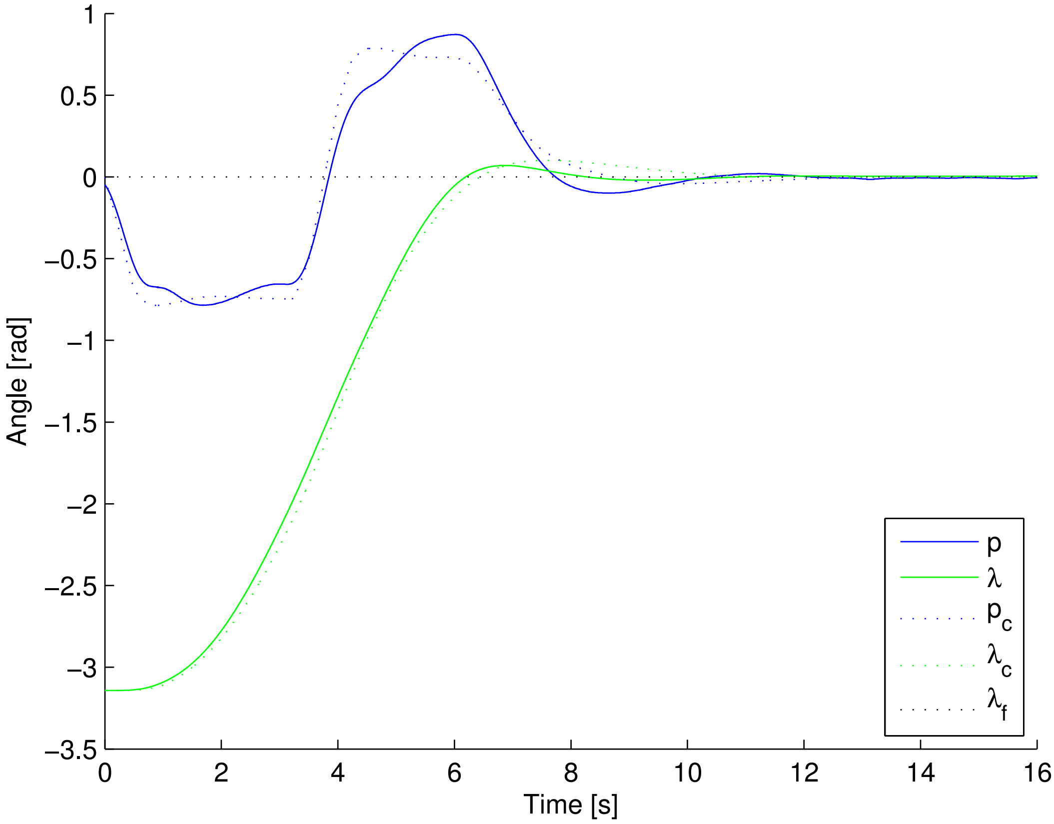

Figure 3: LQR

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

Refer to system id TF for elevation, and we get the following extended continuous state space model:

6.1 State space model

We wish to calculate an optimal trajectory in two dimension, adding a constraint to the elevation. Adding e and $\dot{e}$ to the previous state space model (3), we get

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.03 & -0.39 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -7.13 & -3.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -3.03 & -2.44 \end{bmatrix}}_{A_c} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \\ e \\ \dot{e} \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6.74 & 0 \\ 0 & 0 \\ 0 & 3.13 \end{bmatrix}}_{B_c} \begin{bmatrix} p_c \\ e_c \end{bmatrix} \tag{12}$$

Discuss decoupling...?

6.2 Discretization

Now let $x = \begin{bmatrix} \lambda & \dot{\lambda} & p & \dot{p} & e & \dot{e} \end{bmatrix}^\top$, $u = \begin{bmatrix} p_c & e_c \end{bmatrix}^\top$. Again, using approximate discretization via Euler we obtain a discrete state space model

$$\begin{aligned} x_{k+1} &= (I + \Delta t A_c)x_k + (\Delta t B_c)u_k. && \text{(13a)} \\ &= Ax_k + Bu_k, && \text{(13b)} \end{aligned}$$

where

$$A = \begin{bmatrix} 1 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0.9925 & -0.0975 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.25 & 0 & 0 \\ 0 & 0 & -1.7825 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.25 \\ 0 & 0 & 0 & 0 & -0.7575 & 0.39 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1.685 & 0 \\ 0 & 0 \\ 0 & 0.7825 \end{bmatrix}.$$

6.3 Optimization problem with nonlinear constraints

We calculate an optimal trajectory from $x_0 = \begin{bmatrix} \lambda_0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$ to $x_f = \begin{bmatrix} \lambda_f & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\top$ minimizing the objective function

$$\phi = \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + r_1 p_{c_i}^2 + r_2 e_{c_i}^2, \quad r_1, r_2 \geq 0,$$

or alternatively

$$\phi = \sum_{i=0}^{N-1} (x_{i+1} - x_f)^\top Q (x_{i+1} - x_f) + u_i^\top R u_i, \tag{14}$$

where

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}. \tag{15}$$

The second weighting parameter r_2 is added as we impose an inequality constraint on the elevation for every time step:

$$c(x_k) = \alpha \exp\left(-\beta \left(\lambda_k - \lambda_t\right)^2\right) - e_k \leq 0, \quad k = \{1, \ldots, N\}, \tag{16}$$

where we let $\alpha = 0.2$, $\beta = 20$, $\lambda_t = \frac{2\pi}{3}$.

The objective function (14) is subject to the system dynamics (13a) and thus imposed to linear equality constraints identically defined to that of (8). Similarly to (9) we define the optimization variable z and the matrix G, and the resulting optimization problem can be stated:

$$\min_z \quad z^T G z \tag{17a}$$

subject to

$$A_{eq} z = B_{eq}, \tag{17b}$$

$$c(x_k) \leq 0, \quad k = \{1, \ldots N\}, \tag{17c}$$

$$p^{\text{low}} \leq p_k \leq p^{\text{high}}, \quad k = \{1, \ldots N\}. \tag{17d}$$

6.4 Optional: Additonal constraints

For some clever reason or another we impose additional lower and upper bounds on the elevation rate and travel rate with favourable results...

$$|\dot{e_k}| \leq 0.05,$$
$$|\dot{\lambda_k}| \leq 0.5.$$

6.5 Results and discussion

Because of the non-linearity of (16) it is no longer viable to use a QP-solver, and (17) is solved using MATLAB's `fmincon`: which algorithm (int. point, sqp, active set, trust region) was used...

The optimal input sequence u^* is applied to the plant in an open loop with results show in fig. X. Explain 25 vs. 45 discrepency. We further employ an LQ-controller similarily to 5.1 to guide the trajectory to the set-point and elminiate steady state deviations.

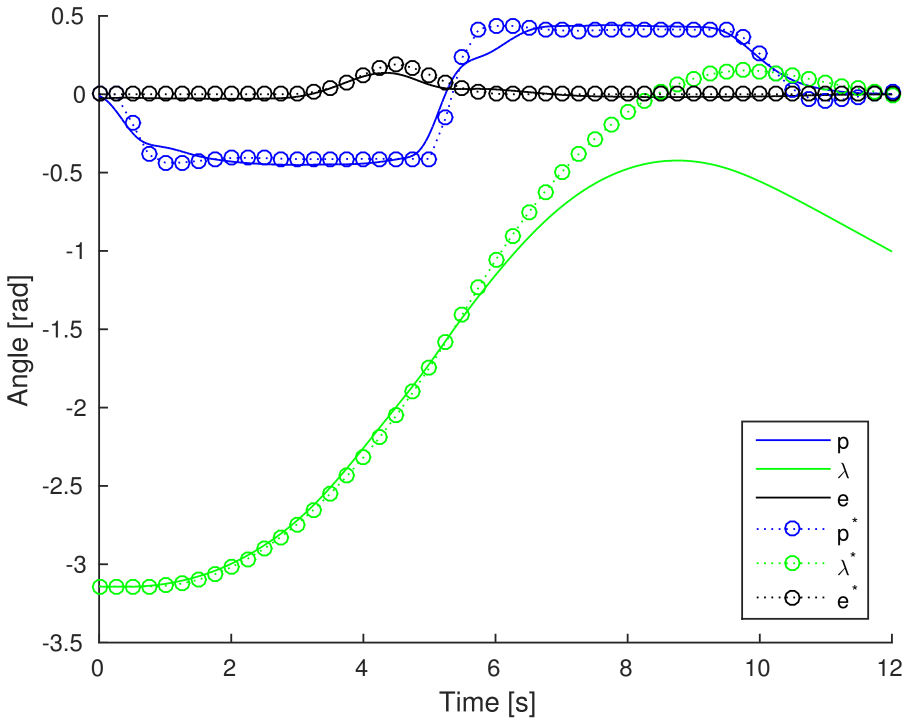

Figure 4: Open Loop 25.

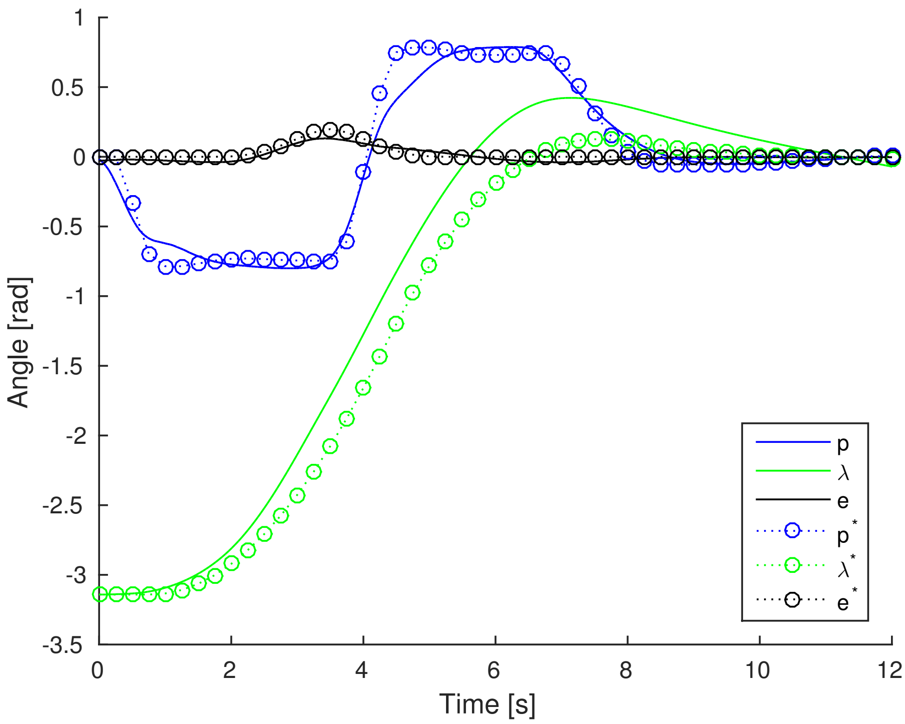

Figure 5: Open Loop 45.

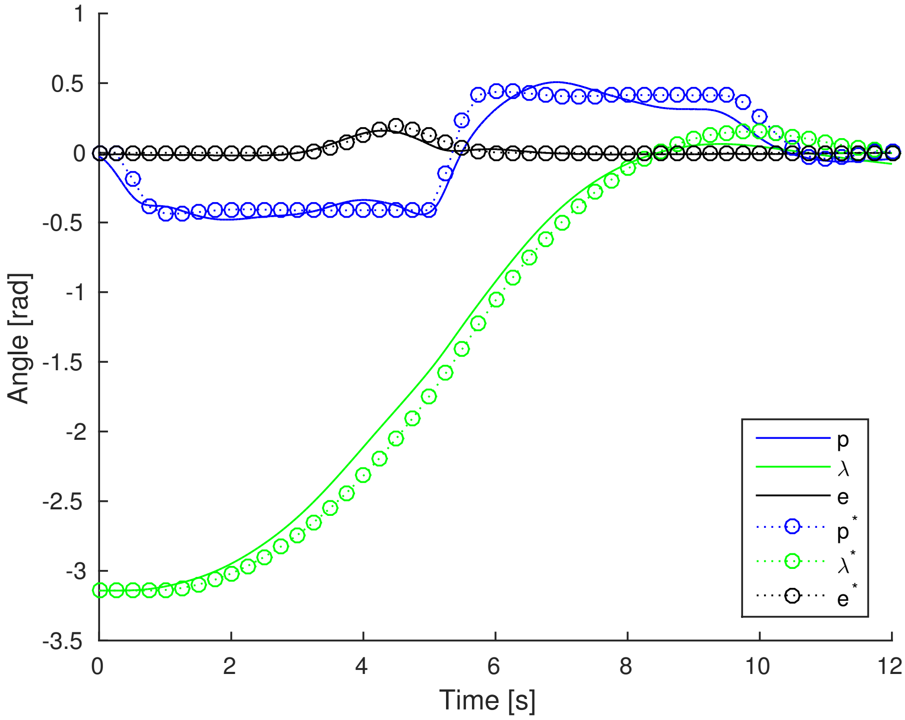

Figure 6: Closed Loop 25.

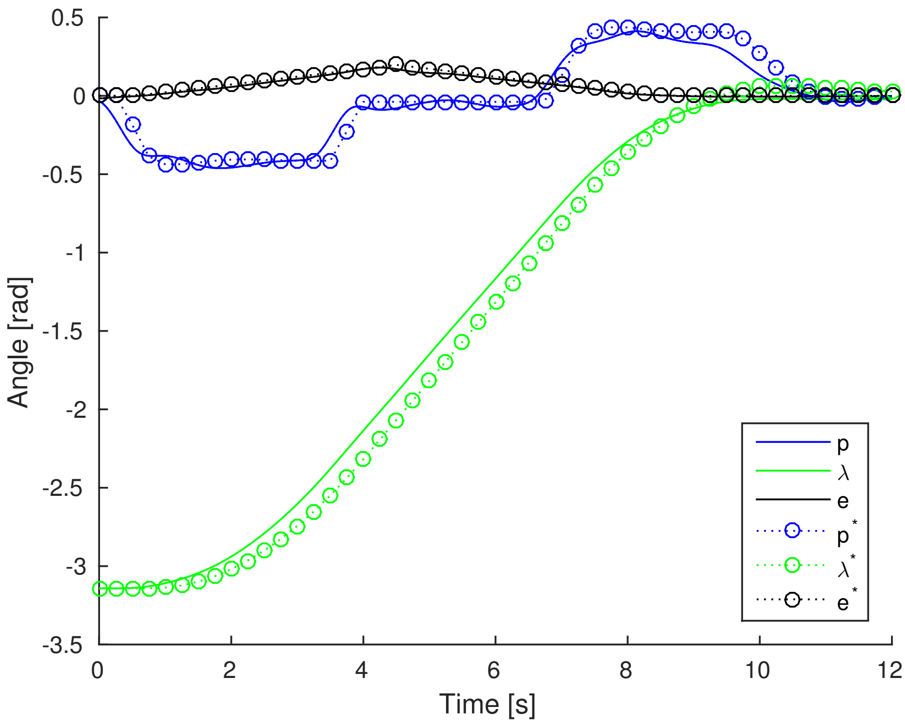

Figure 7: Closed Loop 25, extra constraints.

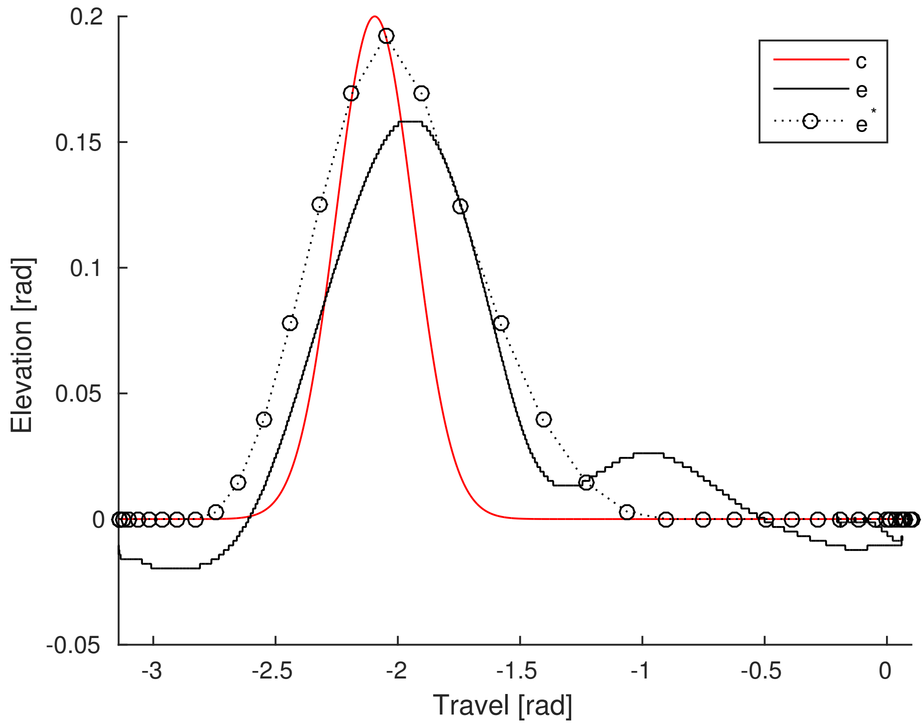

Figure 8: Closed Loop 25 - contraint hill.

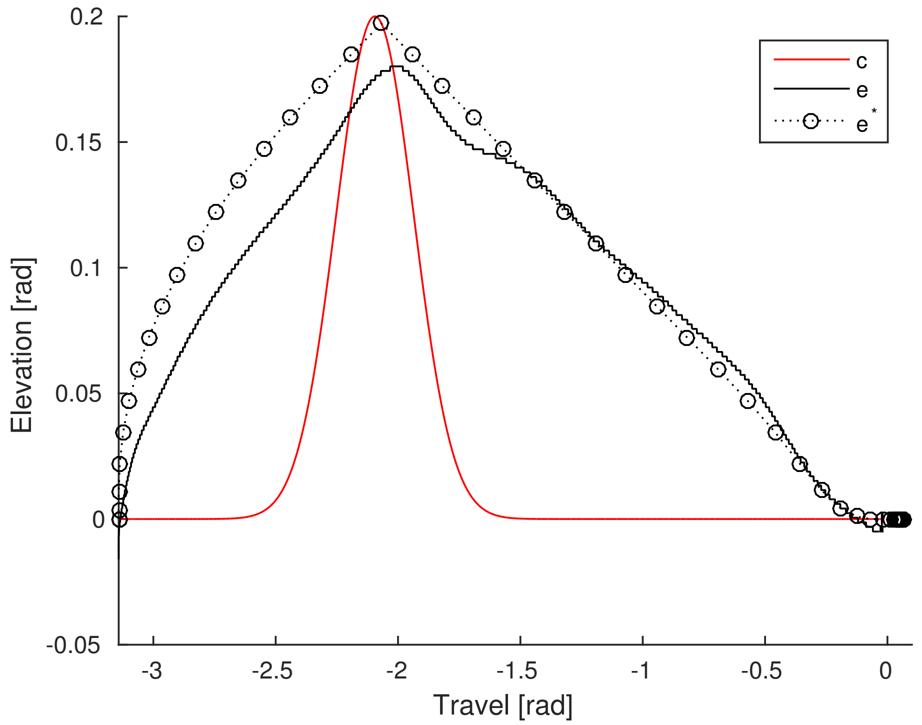

Figure 9: Closed Loop 25, extra constraints - contraint hill.

7 Discussion

A section like this does not have to be long, but write a few short paragraphs that show you understand what you have been doing and how the different results relate to each other.

8 Conclusion

Again, this does not have to be long, but try to write a few reasonable closing remarks.

A MATLAB Code

A.1 System identification

```
keywordstyle

clc
close all

%%%% Pitch %%%%
% h = pitch/pitch_ref

load pitchStep40deg
u_pitch = 40*pi/180 * ones(3000,1);
u_pitch(1) = 0;
y_pitch = pitchStep40deg.signals.values(1:3000);
pitch_data = iddata(y_pitch, u_pitch, 0.001);
pitch_time = 0.001:0.001:3;

opt = tfestOptions('InitialCondition', 'zero');
pitch_sys = tfest(pitch_data, 2, 0,opt); % poles, zeroes

subplot(131);
plot(pitch_time, y_pitch/(40*pi/180), 'r'); % scale for easy comparison with step()
hold on
step(pitch_sys);
hold off
title('Pitch');


%%%% Elevation %%%%
% h = elevation/elevation_ref

load elevStep30deg
u_elev = 30*pi/180 * ones(4000,1);
u_elev(1) = 0;
y_elev = elevStep30deg.signals.values(1:4000) + 16.8*pi/180; % unbias the step
elev_data = iddata(y_elev, u_elev, 0.001);
elev_time = 0.001:0.001:4;

opt = tfestOptions('InitialCondition', 'zero');
elev_sys = tfest(elev_data, 3, 0,opt); % poles, zeroes

subplot(132);
plot(elev_time, y_elev/(30*pi/180), 'r'); % scale for easy comparison with step()
hold on
step(elev_sys);
hold off
title('Elevation');



%%%% TravelRate %%%%
% h = travelRate/pitch

load travelRateStep20deg
travelRate_time = 0.001:0.001:8;
u_travelRate = 20*pi/180 * step(pitch_sys,travelRate_time);
y_travelRate = travelRateStep20deg.signals.values(1:8000) - 0.02; % unbias
```

```
travelRate_data = iddata(y_travelRate, u_travelRate, 0.001);

opt = tfestOptions('InitMethod','iv','InitialCondition', 'zero');
travelRate_sys = tfest(travelRate_data, 1, 0, opt); % poles, zeroes

subplot(133);
plot(travelRate_time, y_travelRate/(20*pi/180), 'r'); % scale for easy comparison with step()
hold on
step(travelRate_sys*pitch_sys, travelRate_time);
hold off
title('Travel rate');
```

B Simulink Diagrams

HIL Initialize block omitted for clarity.

B.1 Open loop optimization

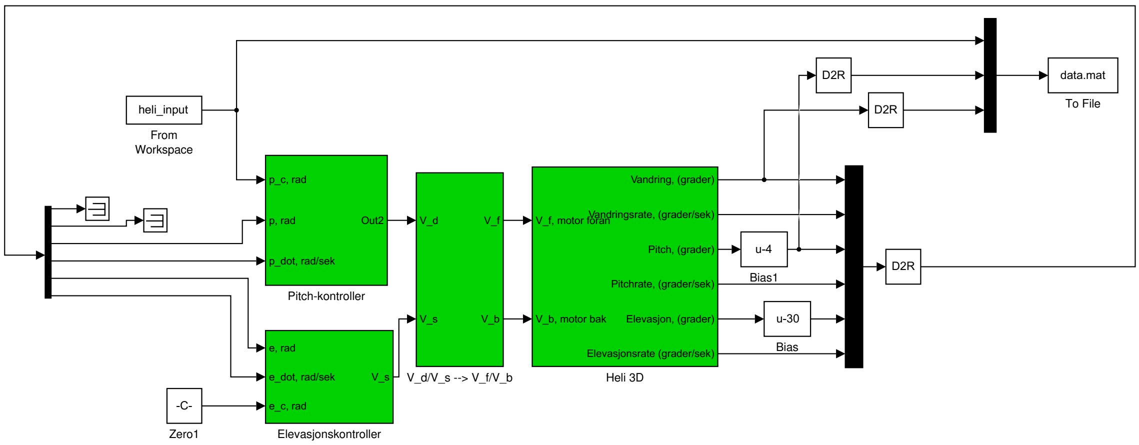

Figure 10: A Simulink diagram.