TTK4135 Optimization and Control
Lab Report

750001, 750002, 750003, and 750004 (Tor Aksel Heirung)
Group 4?

May 4, 2016

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

1 Introduction

In this lab we are going to use the techniques and theory taught in the course of optimization theory and control. To control a small helicopter model. We will be doing this by computing the optimal trajectory of the helicopter and give a corresponding input for this trajectory.

To compute this we need to derive a non-linear model for the dynamics of the system. Then linearize the model around an equilibrium. (add more context)

We will try out a few different setup to measure the performance of the optimal control computed. One where we use the optimal input directly without any feedback. Also we will try to compensate for the inaccuracy of the model by using a feedback controller. In this lab we will be using an linear-quadratic regulator to control the feedback loop.

OLD 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).

You should of course put most of the work into doing good work in the lab and then presenting it in the report. When presenting your work in the report, both content and presentation/layout matters. Since your only way of communicating your good effort in the lab is through writing about it here, the way you write about it is essential. This means that even if you have the very best controller but describe it poorly, you will probably not be rewarded for the good results. A plot showing perfect control is worth very little if it is not accompanied by a clear description of what it represents.

Layout is naturally less important than content, but it still matters. You can think of report writing like selling an apartment; when you present your apartment for potential buyers you will of course clean the apartment and make it good looking. How clean the apartment is does of course not determine its value, but it is still important since it influences the subjective value your buyers will put on the apartment.

1.1 Software

You are of course free to use whatever software you want for report writing. You can also submit a handwritten report, although this is probably not a great idea if your handwriting can be hard to read.

You can also use Word or a similar word processor. However, it is next to impossible to achieve decent layout with Word. The support for vector graphics (discussed later) is extremely poor, and text tends to look pretty bad (bad support for kerning and ligatures). Furthermore, math is both time consuming and difficult to input, and tends to look very ugly. In general, a report written in Word looks like a draft.

It is strongly recommended to use Latex. Unless you tweak the layout too much, your report will almost certainly look very good. Although it can take a bit of effort to get started, it is also much quicker to use than Word and similar programs. The support for math and vector graphics is also great.

If you are new to Latex, you can have a look at the source for this document to get started. You can also look at the presentation by Berland (2010) (in Norwegian) or consult Oetiker et al. (2011). Another good reason to learn Latex is that you probably don't want to write your master's thesis in something like Word, doing so would likely be very frustrating. Being reasonably fluent in Latex before you get that far will make your thesis work much smoother.

Some of you are probably fluent in Latex and might plan to write the report using it. Please resist the temptation (if any) to change the fonts, make super fancy headers (they are not necessary for a report like this), change the margins, change the paragraph indentation and/or spacing, and similar things.

A great tool for collaborating on Latex documents is ShareLaTeX at `https://www.sharelatex.com/`; if you use this you won't have to install anything on your computer. Texmaker at `http://www.xm1math.net/texmaker/` is a good cross-platform editor. Some people like Lyx, which is a Latex editor that behaves a little bit like Word.

1.2 Other Comments

If you have problems with Latex, the solution is usually just a few Google searches away.

You can write the report in Norwegian or English. Writing in English is encouraged and is great practice, but entirely optional. *Do not interpret any of the advice or suggestions here as requirements.*

This report is organized as follows: Section 2 contains a few remarks on report writing and some random Latex advice. An example of a table can be foundin Section 3, along with two remarks on report writing. Section 4 contains some advice on using plots from MATLAB. A few suggestions for making illustrations are given in Section 5; a matrix equation can also be found here. Section 6 has a few comments on references and floats in Latex. The closing discussion and concluding remarks are in Sections 7 and 8, respectively. Appendix A contains a MATLAB file while Appendix B shows an example Simulink diagram. The Bibliography can be found at the end, on page 45.

2 Problem Description

2.1 Lab Setup

The helicopter, as shown on figure 1, is constructed from two main parts.
The basis and the arm. The arm has got on one side two propellers and on
the opposite side a counter weight. The arm has got 2 degrees of freedom
and can therefore move up and down. The two propellers are attached to
the arm by a rotary bound, as shows figure **??**. The body of the helicopter
can also rotate around its axis. Combined with the movement of the arm
we observe the **travel**. The rotation of the propellers around the arm is
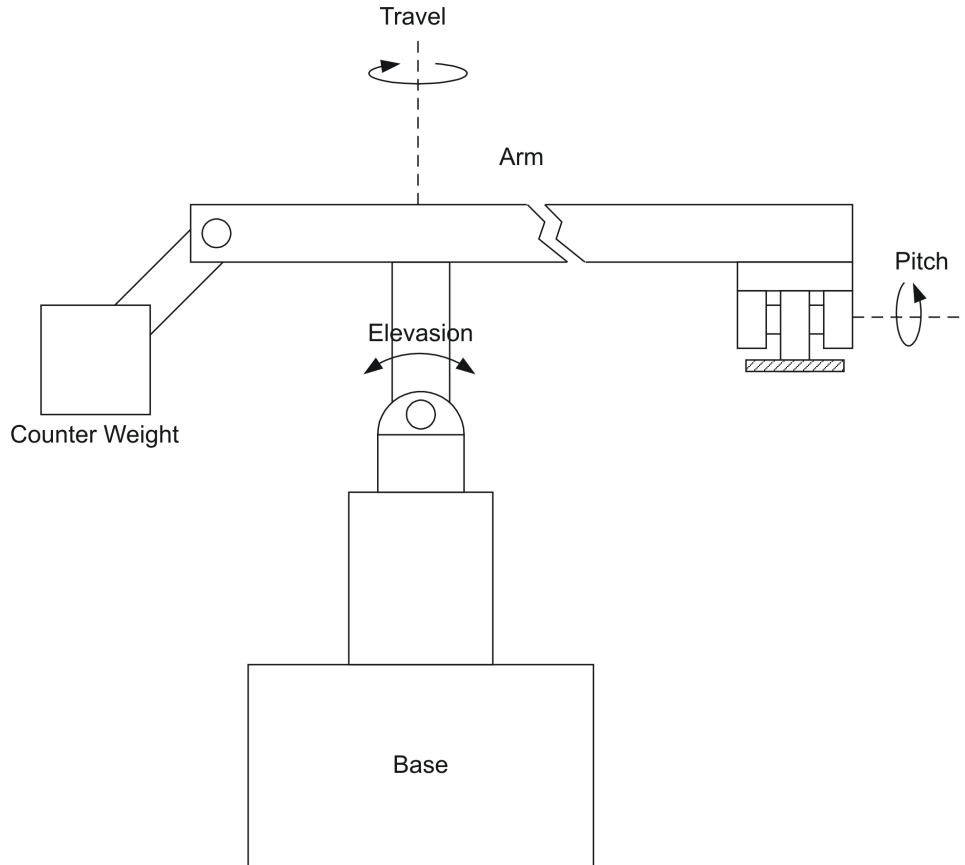denoted as the **pitch**.

Figure 1: the elevation of the helicopter

The model is given by the quations (**??**). Equation (1a) describes the
elevation, equation (1b) accesses the pitch angle. The speed is the derivation
of the of the path as described in equation (1c). The travel accelaration is
given by equation (1d).

$$\ddot{e} + K_3 K_{ed}\dot{e} + K_3 K_{ep}e = K_3 K_{ep}e_c \tag{1a}$$

$$\ddot{p} + K_1 K_{pd}\dot{p} + K_1 K_{pp}p = K_1 K_{pp}p_c \tag{1b}$$

$$\dot{\lambda} = r \tag{1c}$$

$$\dot{r} = -K_2 p \tag{1d}$$

These equations were derived from:

$$J_2\ddot{e} = l_a K_f V_s - T_g \tag{2}$$

s.t.

$$\ddot{e} = K_3 V_s - \frac{T_g}{J_e}, \ K_3 = \frac{l_a K_f}{J_e}.$$

The model is subsequently discretized into

$$\Delta x_{i+1} = A\Delta x_i + B\Delta u_i$$

where

$$\Delta x = x - x^* \tag{3}$$

$$\Delta u = u - u^*. \tag{4}$$

We want to minimize the cost function

$$\phi = \sum_{i=1}^{N}(\lambda_i - \lambda_f)^2 + qp_{ci}^2, \ q \geq 0 \tag{5}$$

2.2 Introduction to Simulink /QuaRC

Simulink is a program for Model-Based Design. It overtakes the code from Matlab, compiles it into the C language and sends it to the right controllers. QuaRC[1] is a Real-Time control system, that is integrated into Simulink. To control the program, Matlab and Simulink is used. We use QuaRC for the build option as can be shown on figure ??. The work flow is that we first build the program. In Simulink is than the Matlab code compiled to the C language with Visual C++. Then the code is downloaded to QuaRC. We have also to set the following parameters, if we already didn't do so, like buffer size, sampling frequency. After this, the helicopter can be started. We assure that the power button at the helicopter is on and on the computer we can push Start. After the flight, we can compare the expected flight from the real flight in a Matlab figure. The realtime measurments will always be shown up as a piece wise constant function.

---

[1]`http://www.quarcservice.com/ReleaseNotes/files/quarc_user_guide.html`

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	$\mathrm{N\,V^{-1}}$
J_e	Moment of inertia for elevation	0.83	$\mathrm{kg\,m^2}$
J_t	Moment of inertia for travel	0.83	$\mathrm{kg\,m^2}$
J_p	Moment of inertia for pitch	0.034	$\mathrm{kg\,m^2}$
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

3 Repetition/Introduction to Simulink/QuaRC

This section should not be very long.

If you want, you can use the source for Table 1 to see how a (floating) table is made.

Variables and symbols are always in italics, while units are not.

4 Optimal Control of Pitch/Travel without Feedback

Answer all the parts of the first exercise in an organized and clear manner. You should of course try to get good results in all the exercises, but if you have made a good effort without achieving great performance, a good discussion of possible reasons is just as good. Present your thinking and efforts and discuss possible reasons for good or bad results.

Include plots of all relevant results, but make sure you don't overwhelm the reader with too many plots. Also keep in mind that the plots should be as "readable" as possible; that is, they should not be too hard to interpret.

There are some important things to consider when exporting figures from MATLAB, most importantly which format you use. Never ever use JPEG for anything that is not a photography or similar. Any figure, like a plot or block diagram, must never be stored as a JPEG. If you zoom in on Figure 2 you can see a lot of noise close to any of the dark curves and lines, this is due to the compression in JPEG. Figure 2 will look horrible both on a screen and on paper.

The PNG format is slightly better for plots, but since it is a raster format (a grid of pixels), it looks ugly if you zoom in. It also looks ugly if you scale it, both on a screen and on paper. Try to avoid PNG if you can. Figures 3 and 4 are both PNG figures; the latter being a larger figure scaled more than the former. Note both how choppy and ugly the blue curve is, and how the different sizes create inconsistent font sizes.

The simplest way to get a reasonably good looking plot is to save it as EPS in MATLAB. Do this by clicking "File" in the figure window, and the "Save As. . . "; choose "EPS file (*.eps)" in the "Save as type:" menu.[2] Figure 5 shows a plot in EPS format. Since EPS is a vector format, the Figure can be scaled and still look good (but mind the font size!). If you zoom in you can see that the curve and the letters/numbers are smooth. A figure in vector format will usually look good both on a screen and on paper.

Note that the size of the actual figure window in MATLAB determines how large the exported figure is. Hence, if you enlarge the figure window before exporting, you will need to scale the figure by a larger factor in the report. This will lead to a tiny font in the figure. There are many better ways of exporting graphics from MATLAB, but they quickly become fairly involved. The above method of exporting to EPS will in most cases give nice figures.

You can write Latex in your MATLAB figures. The script used to create Figures 2–5 is included in Section **??**. Do not use a screen shot of a scope of figure in MATLAB in your report.

---

[2] pdfLatex does not support EPS directly, but since we have loaded the *epstodf* package, this is not a problem.
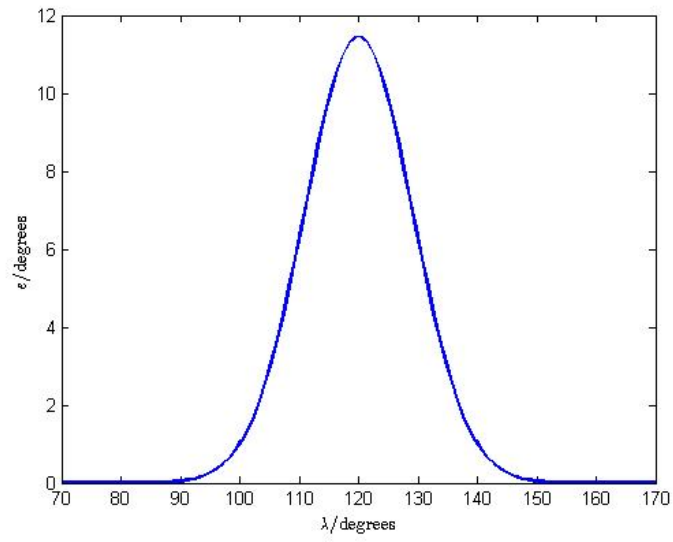
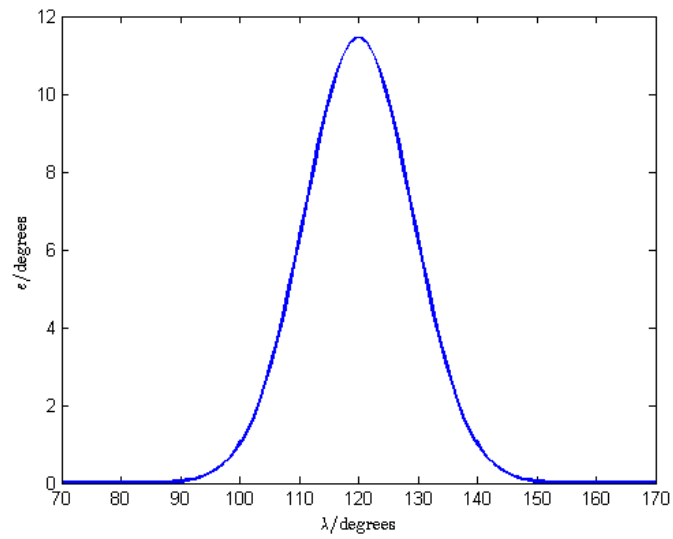Figure 2: A plot in JPEG format — a very bad idea.
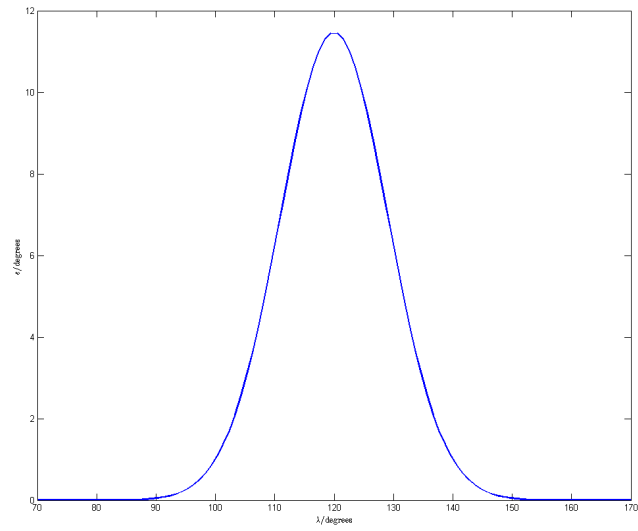

Figure 3: A plot in PNG format — a bad idea.

Figure 4: A plot in PNG format — a bad idea. This figure is originally larger than the other PNG figure, but both are scaled to the same size.
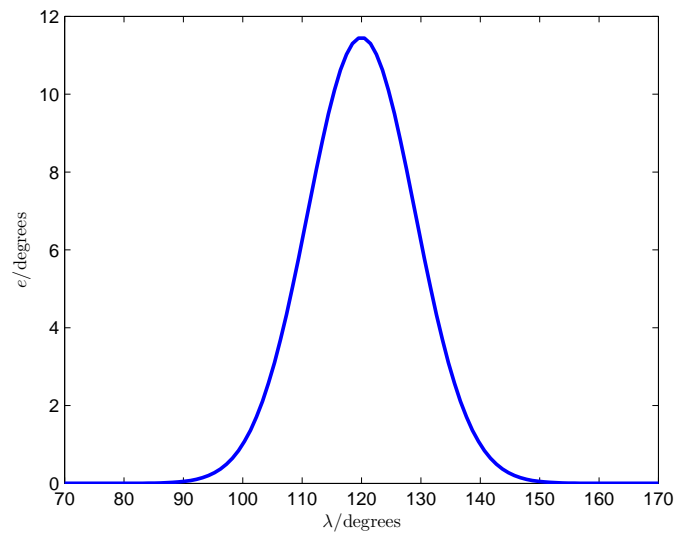

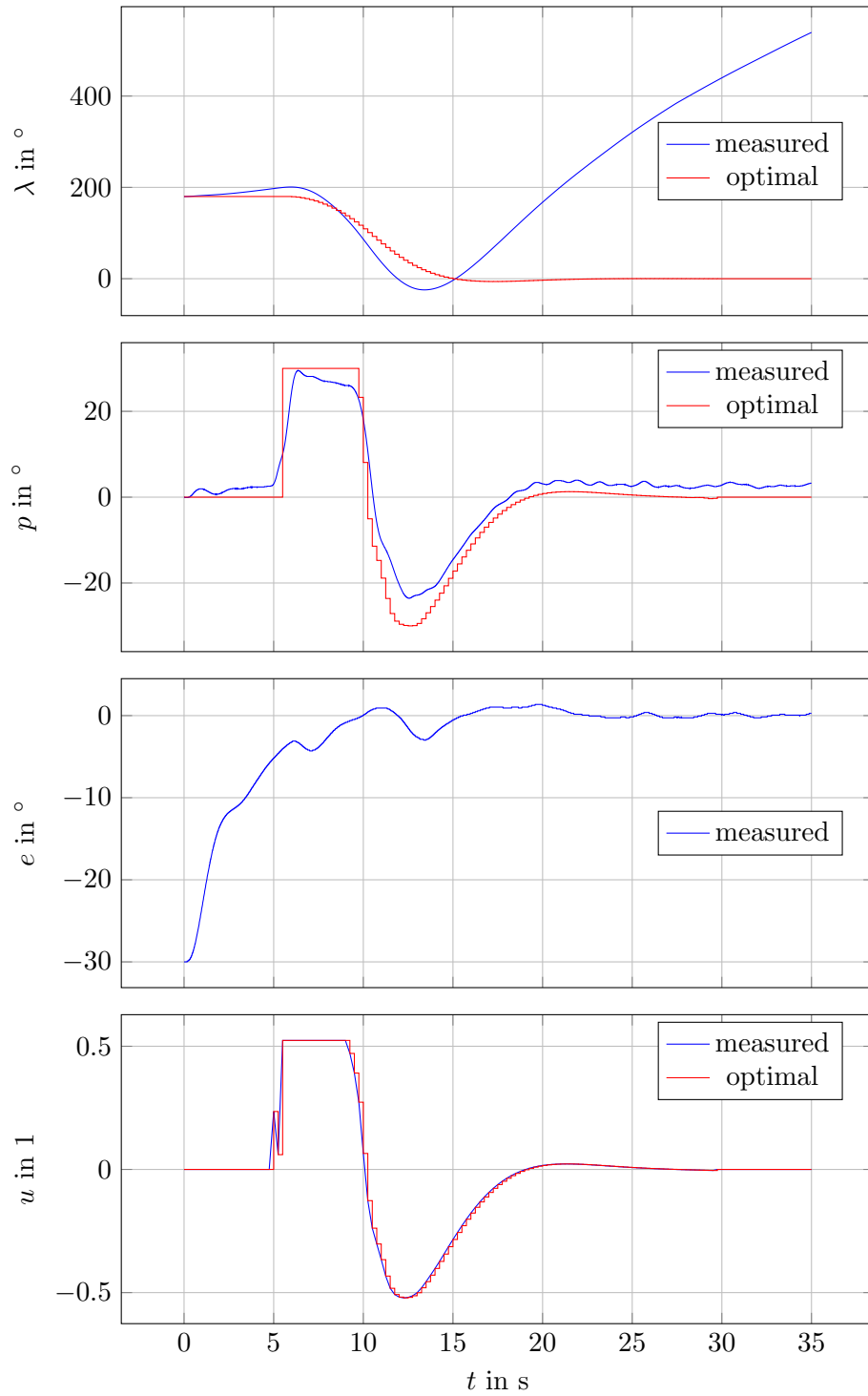Figure 5: A plot in EPS format — a much better idea.
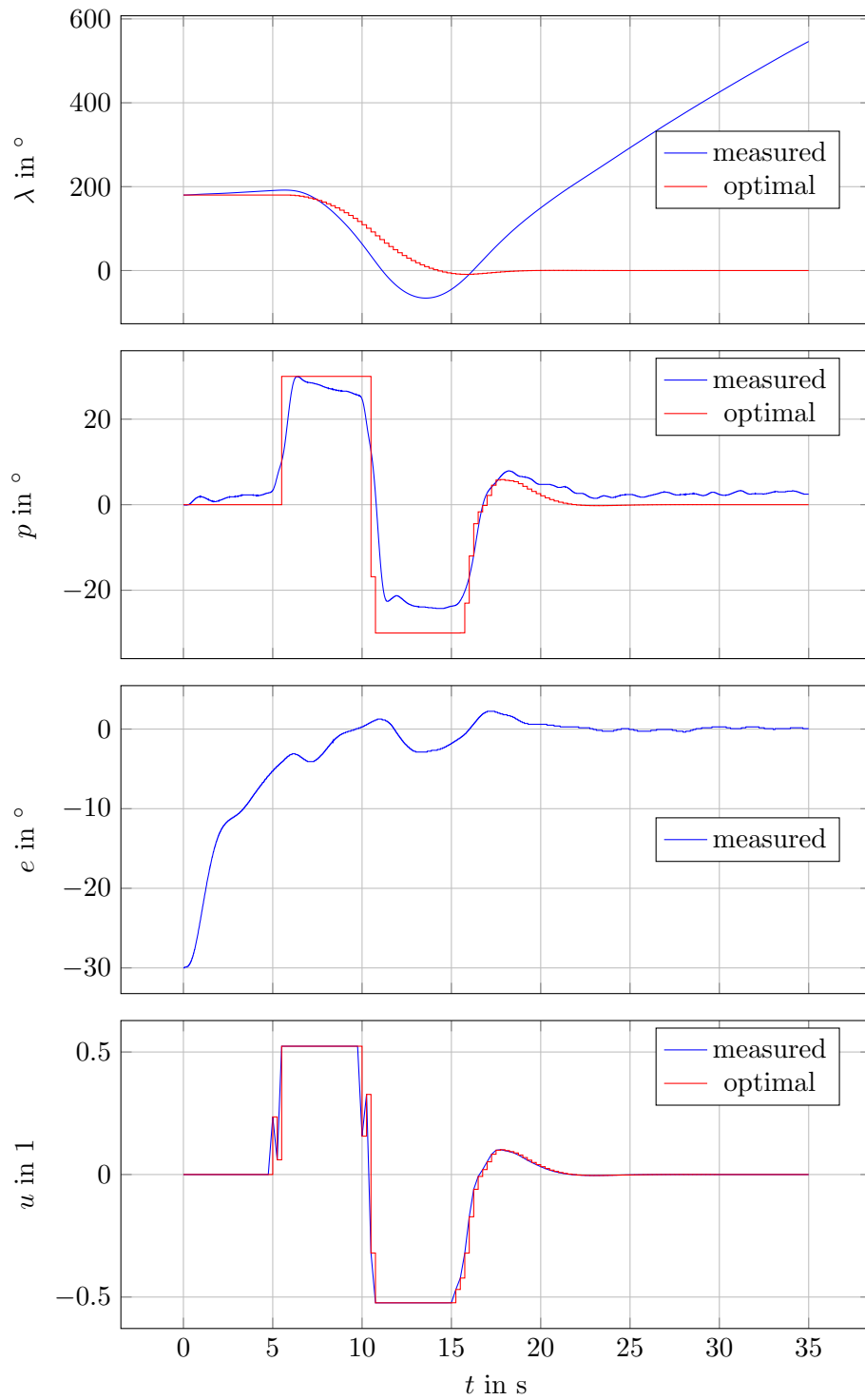
11

Figure 6: problem2plots_q_1.0
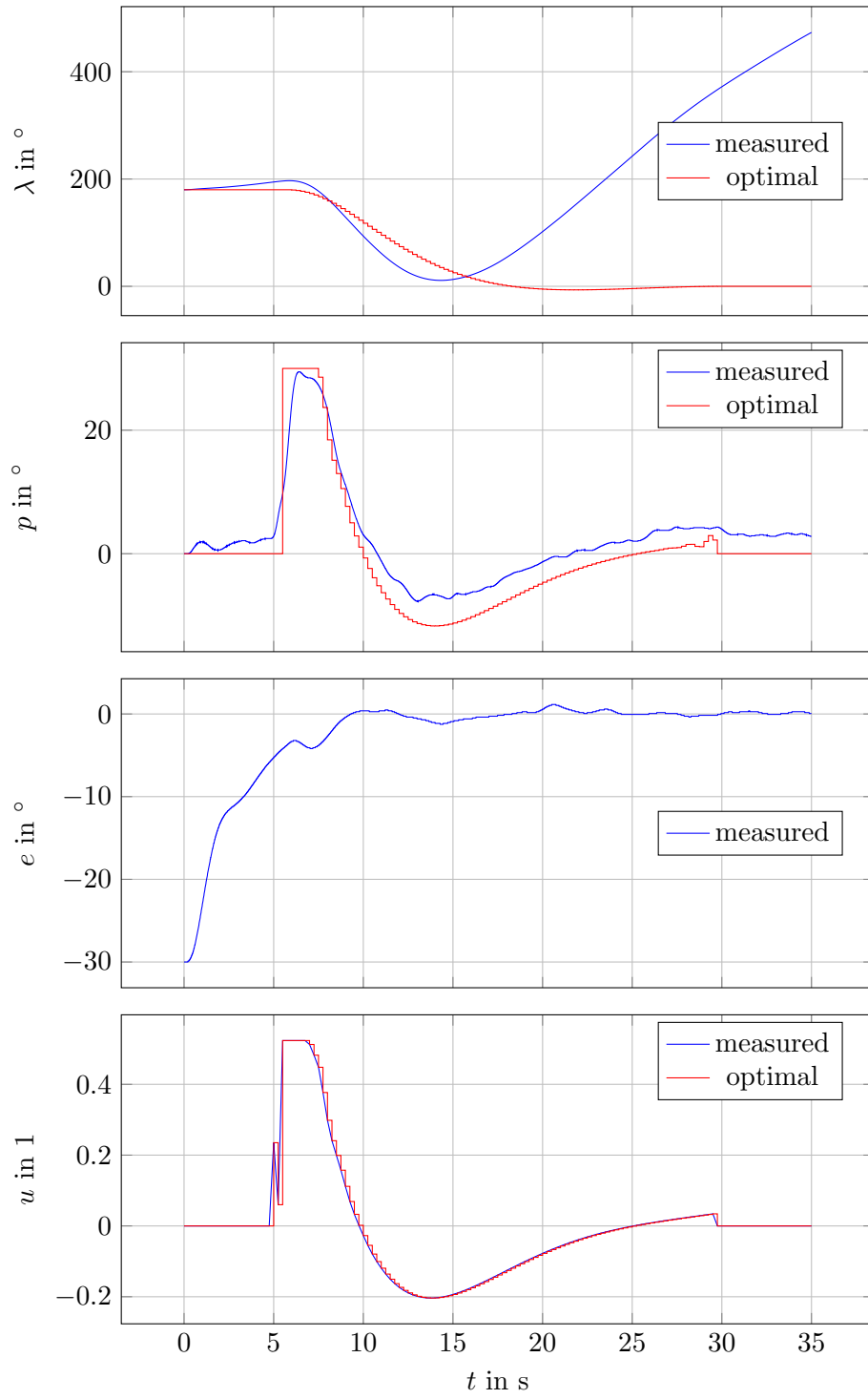
Figure 7: problem2plots_q_0.1
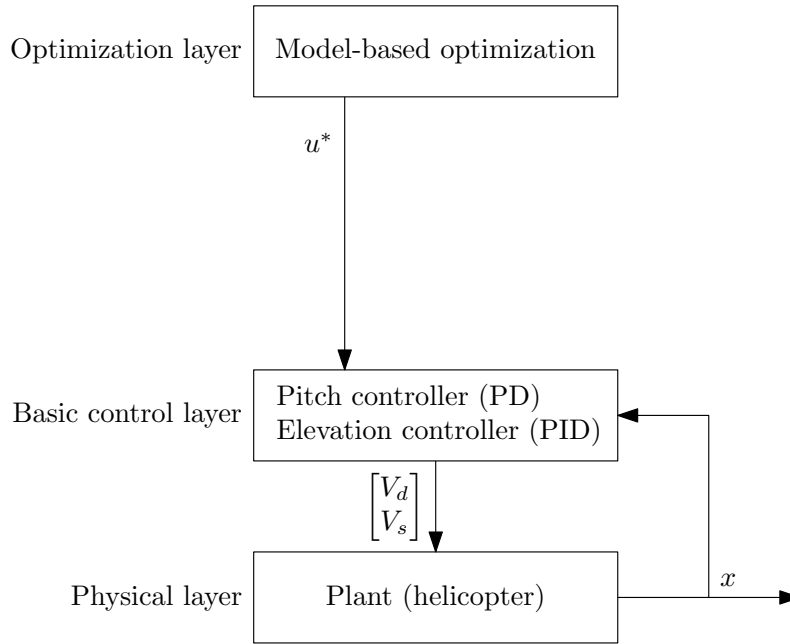
Figure 8: problem2plots_q_10

Figure 9: A figure created with Ipe.

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

You are as mentioned welcome to use the figures from the assignment text if you want to (cite the source!). You can also draw your own (cite the source if it is heavily based on someone else's.). Figure 9 was created quickly with Ipe. Inkscape is a good alternative for more advanced illustrations. Some people prefer the Latex package TikZ (http://texample.net/tikz/examples/), but this takes a little effort to learn.

Here is a matrix equation you can use as a template:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & -b & 0 & 0 & 0 \\
-a & 1 & 0 & 0 & 0 & -b & 0 & 0 \\
0 & -a & 1 & 0 & 0 & 0 & -b & 0 \\
0 & 0 & -a & 1 & 0 & 0 & 0 & -b
\end{bmatrix}
\begin{bmatrix}
x_1 \\ x_2 \\ x_3 \\ x_4 \\ u_0 \\ u_1 \\ u_2 \\ u_3
\end{bmatrix}
=
\begin{bmatrix}
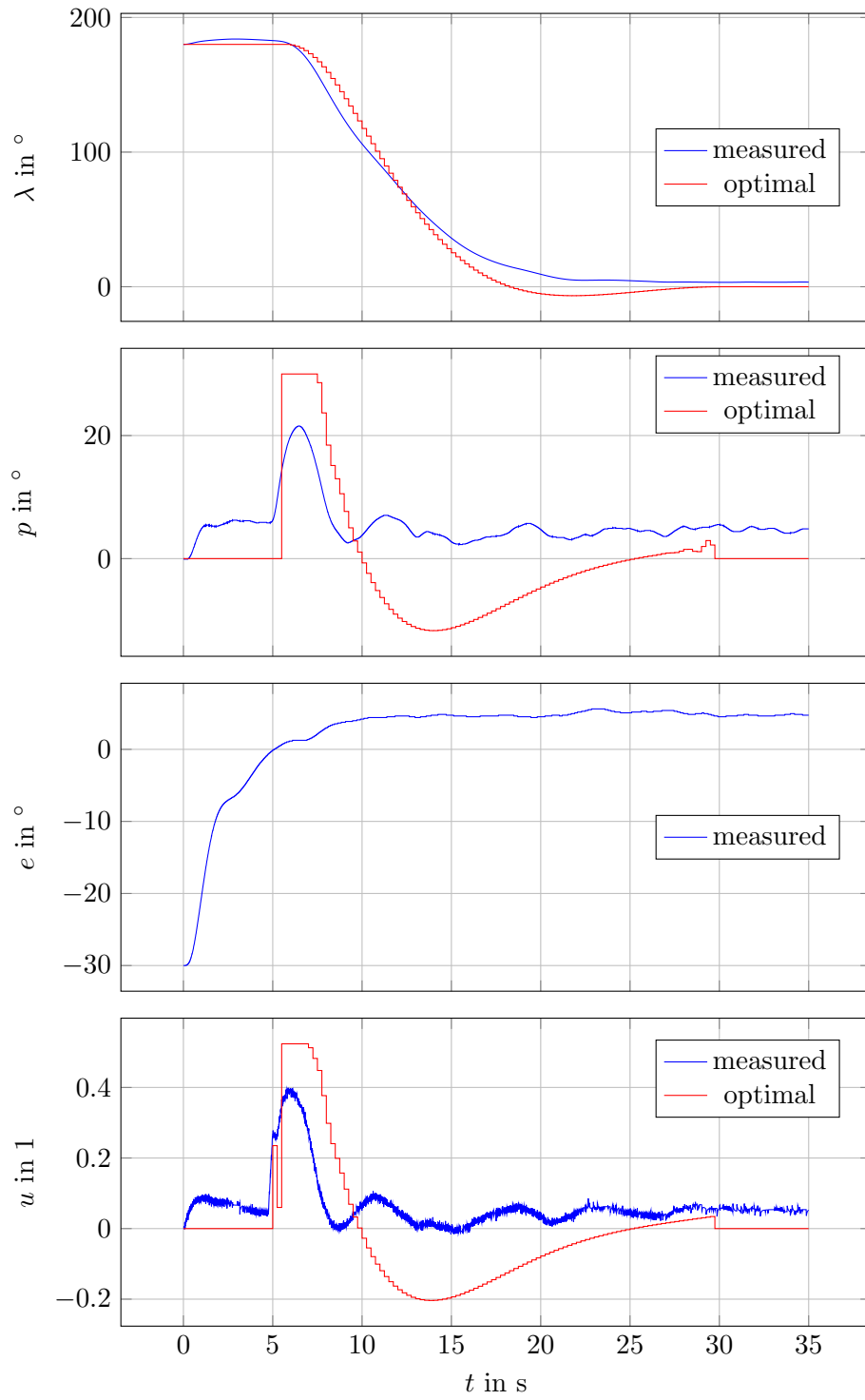ax_0 \\ 0 \\ 0 \\ 0
\end{bmatrix}
\tag{6}
$$

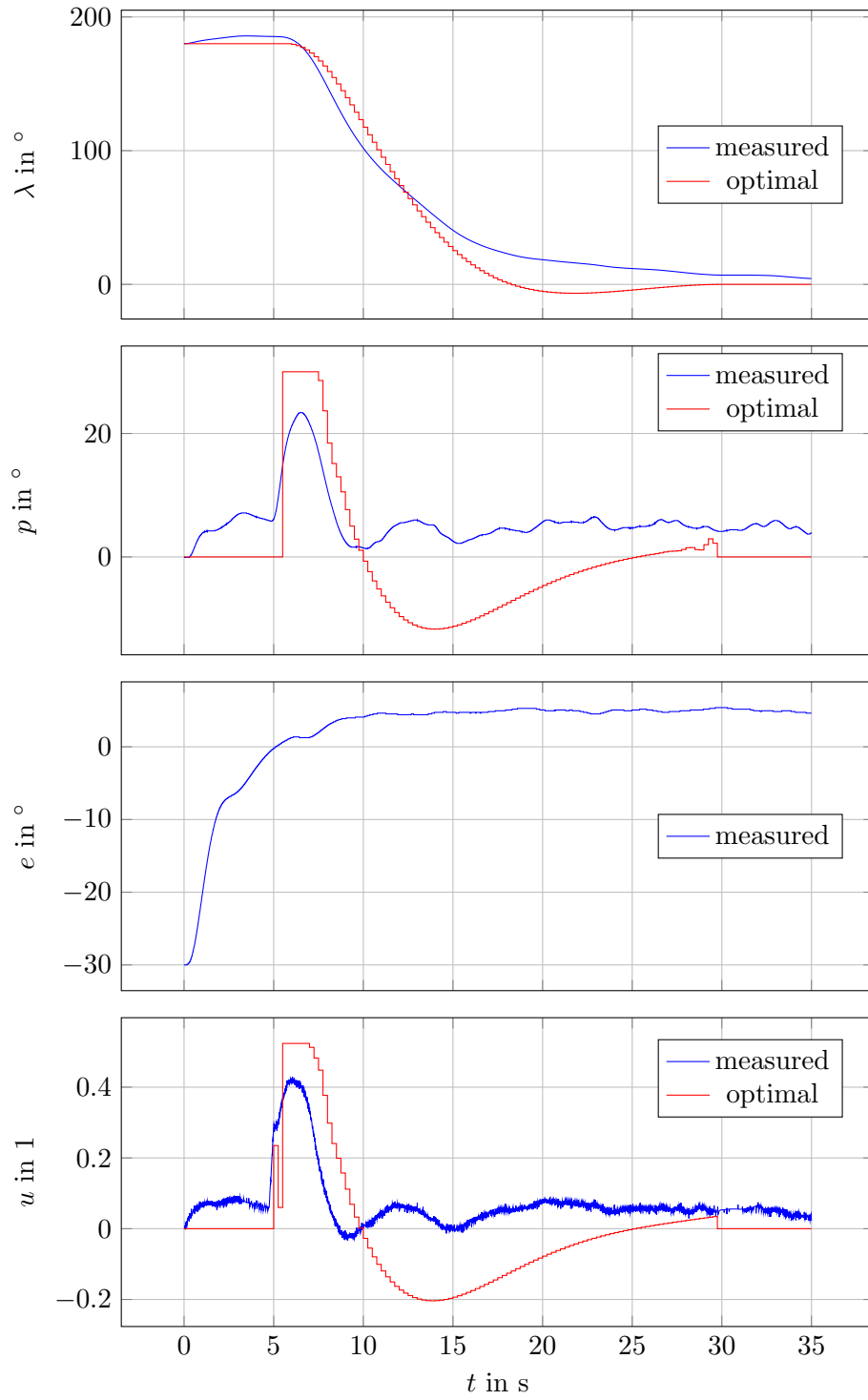Figure 10: problem3plot_1010_1

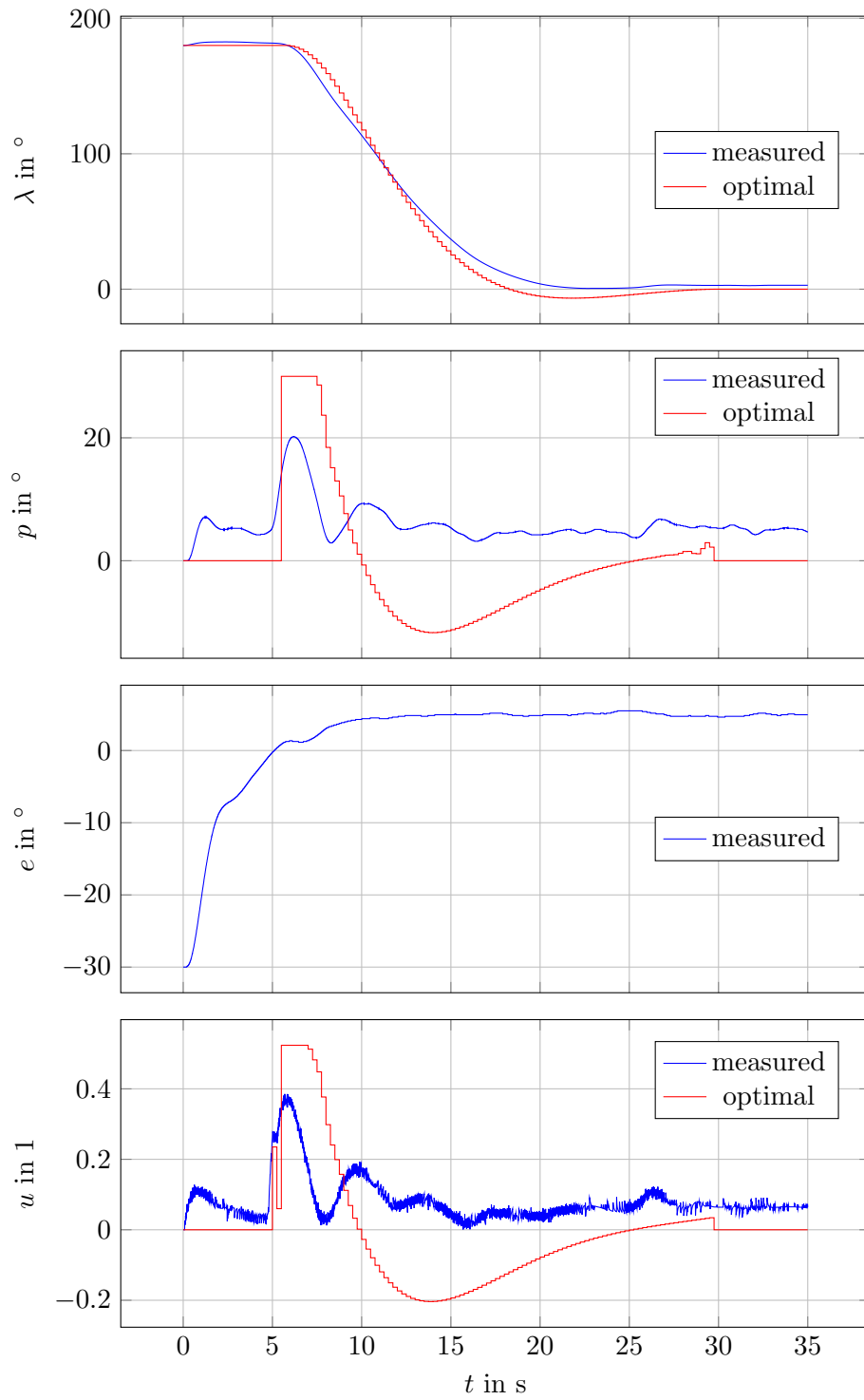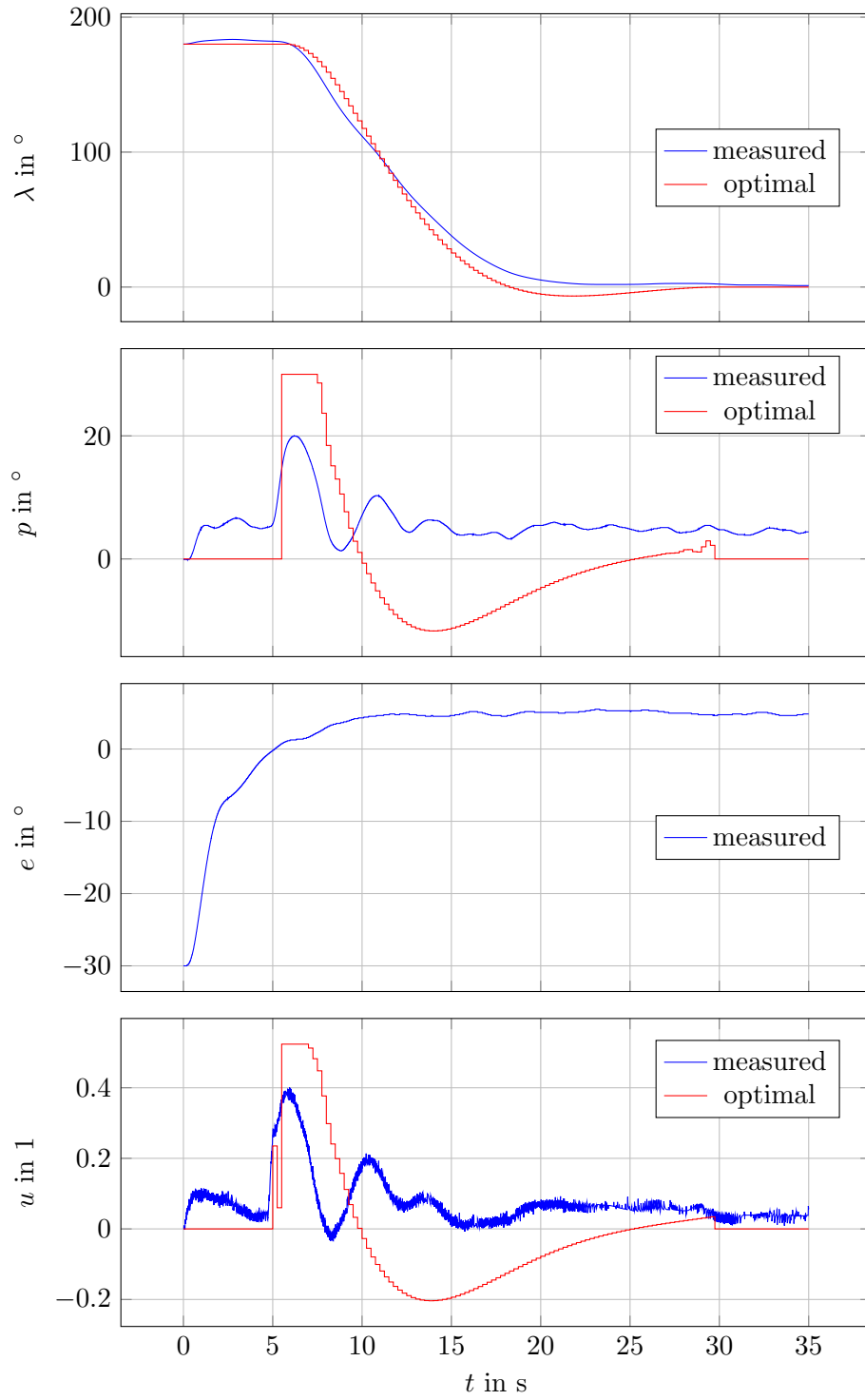Figure 11: problem3plot_1050_1

17

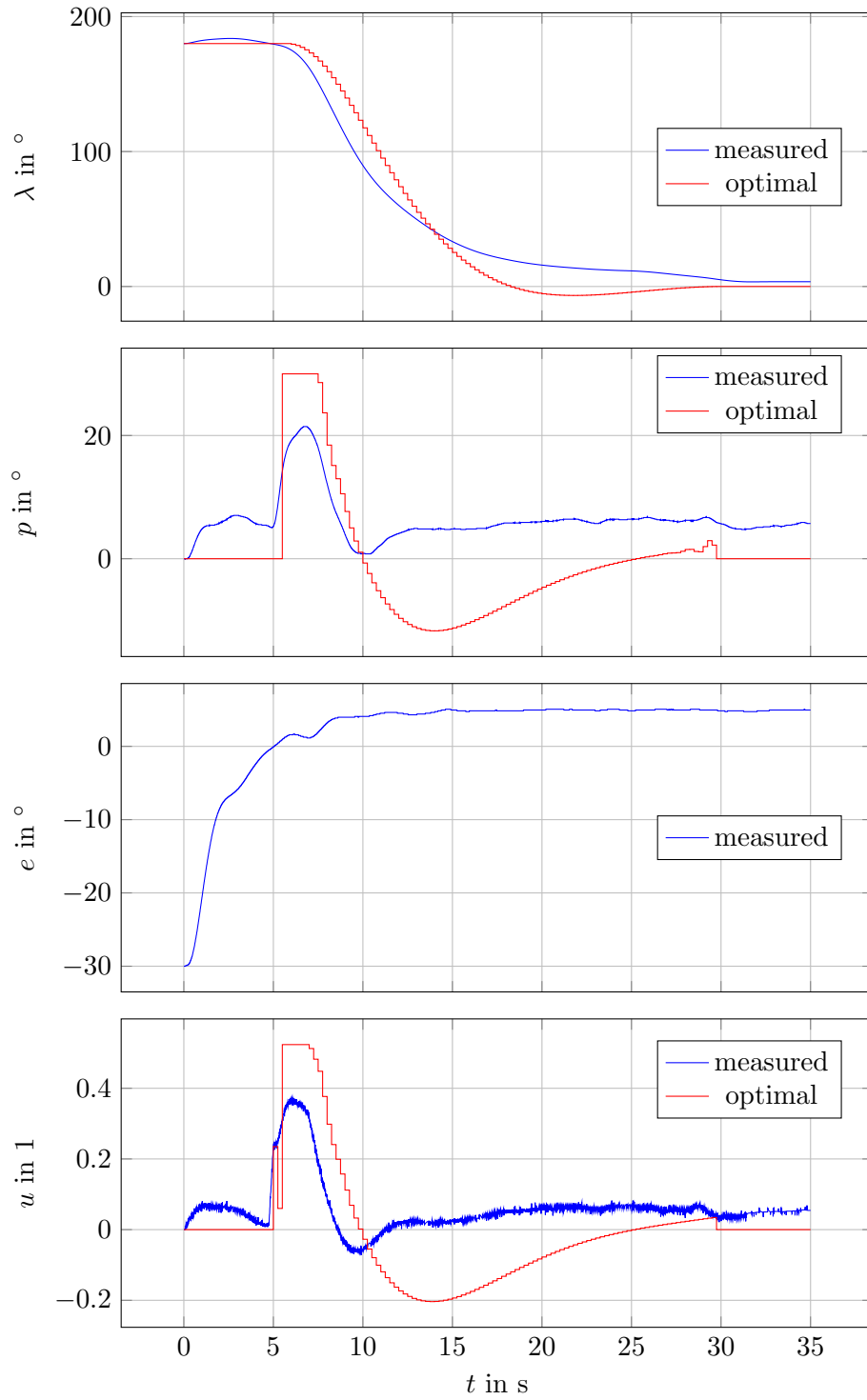Figure 12: problem3plot_5010_1

18

Figure 13: problem3plot_10150_0.5
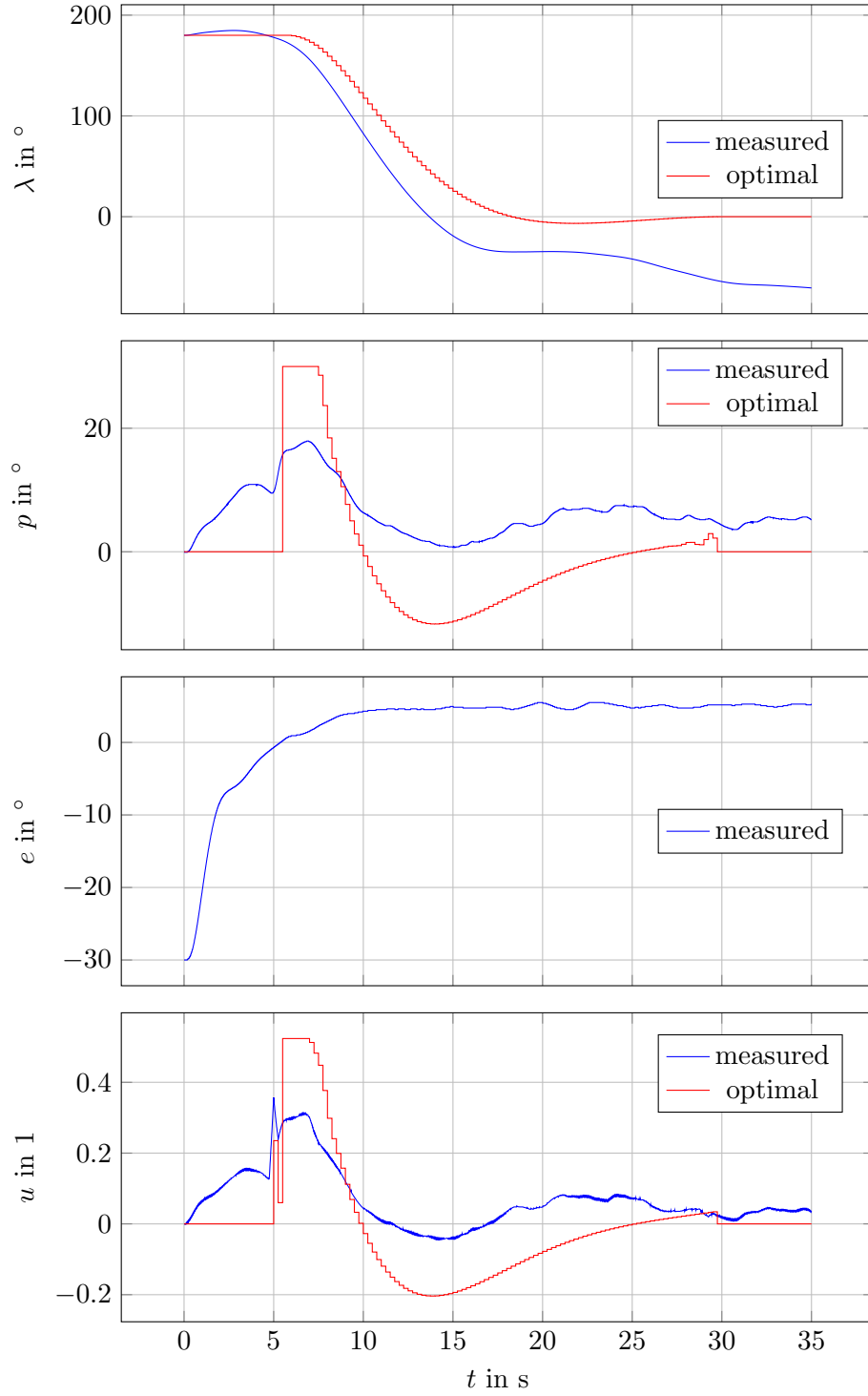
Figure 14: problem3_LQR[1,0,10,0], R = 0.5

Figure 15: problem3_LQR[1,0,10,10], R = 0.5
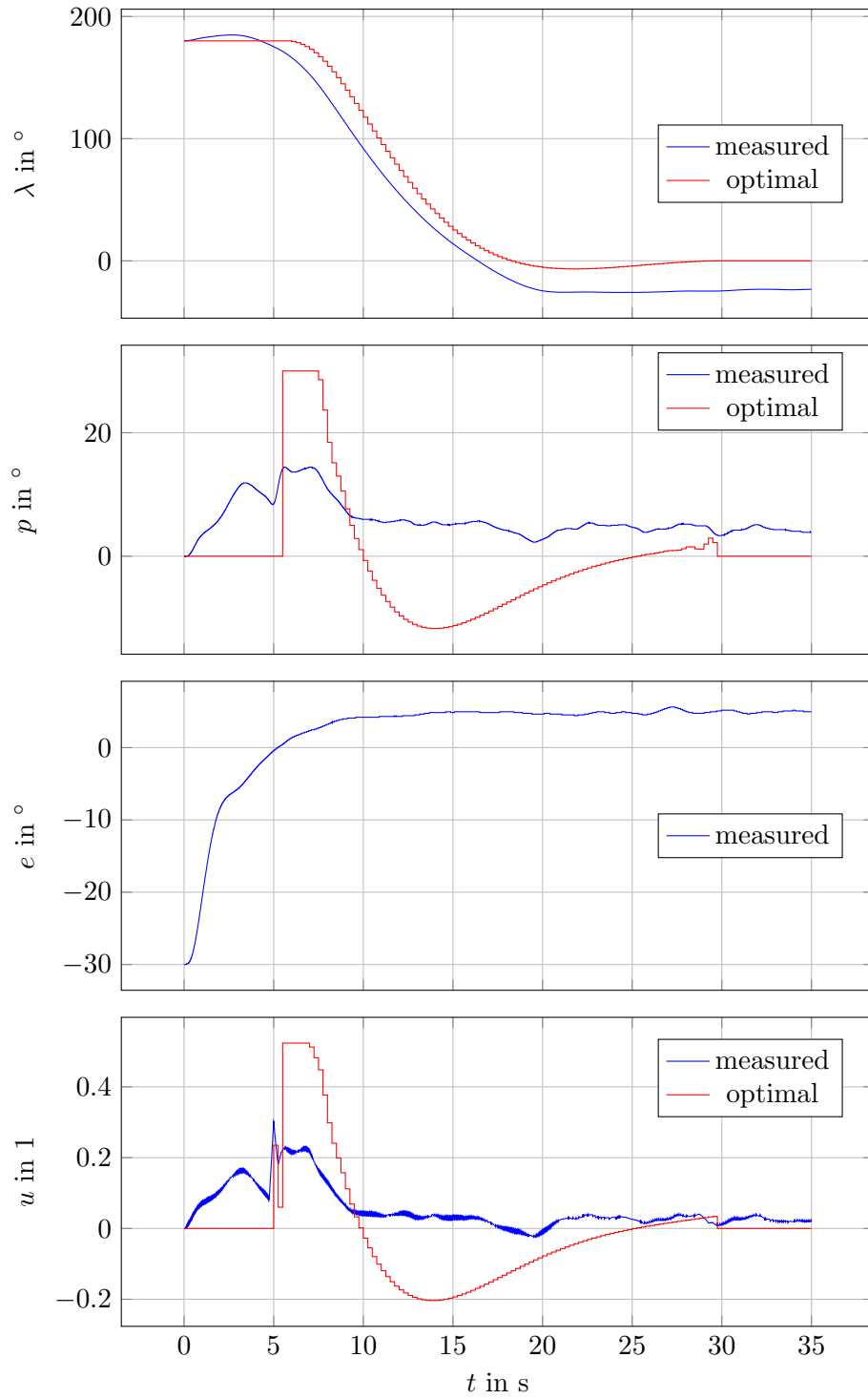
Figure 16: problem3_LQR[5,0,0,5], R = 0.5
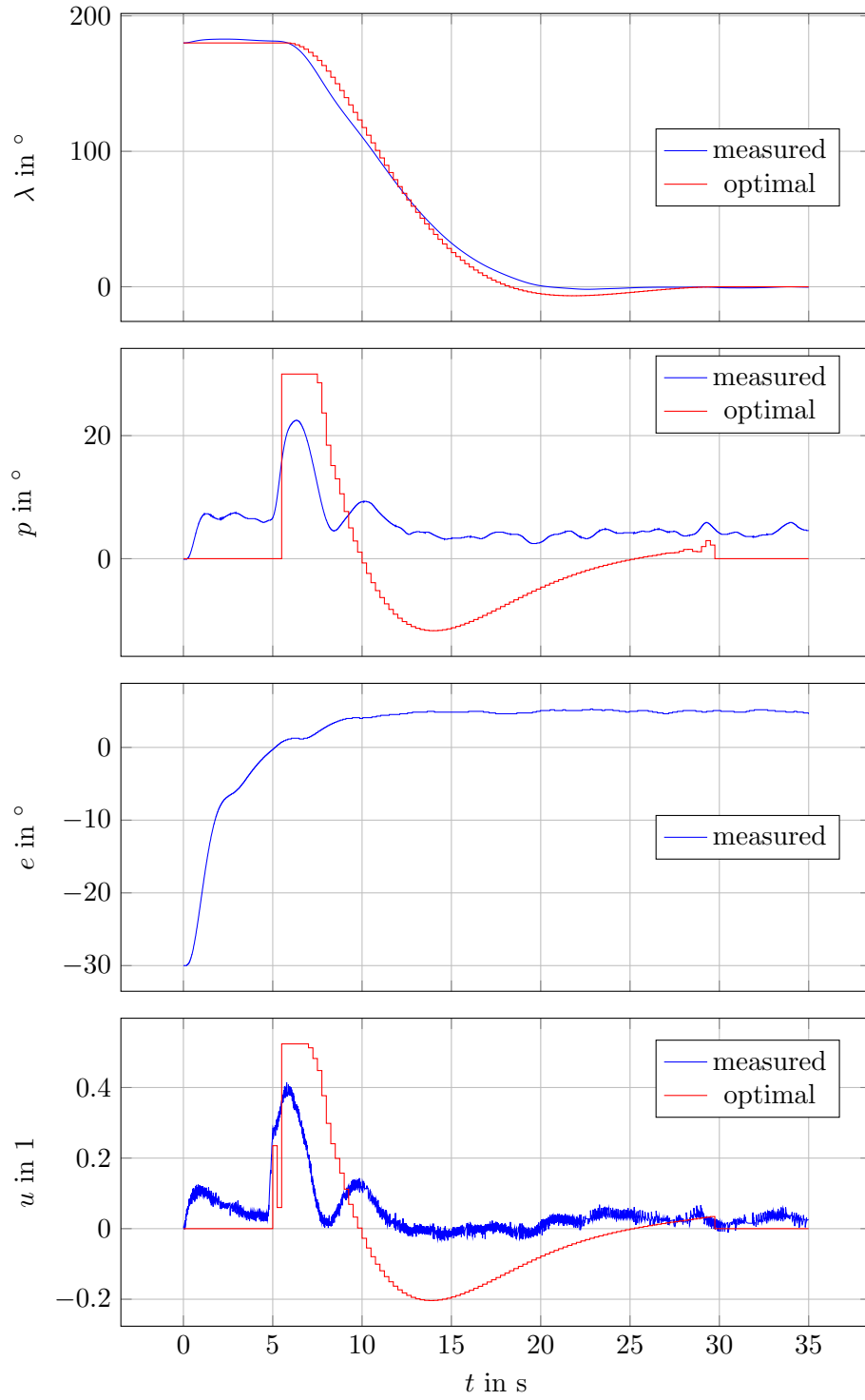
Figure 17: problem3_LQR[10,1,5,0], R = 0.5

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

In this part of the excersice a constraint on the elevation is added. Therefore the equation describing the dynamics of the elevation e from (??) must be added to the state space representation of the model of the helicopter

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{A}_c\mathbf{x} + \mathbf{B}_c\mathbf{u} \tag{7a}$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -K_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -K_1 K_{pp} & -K_1 * K_{pd} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -K_3 * K_{ep} & -K_3 * K_{ed} \end{bmatrix} \tag{7b}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ K_1 K_{pp} & 0 \\ 0 & 0 \\ 0 & K_3 K_{ep} \end{bmatrix} \tag{7c}$$

$$\mathbf{x} = \begin{bmatrix} \lambda & r & p & \dot{p} & e & \dot{e} \end{bmatrix}^T \tag{7d}$$

$$\mathbf{u} = \begin{bmatrix} p_c & e_c \end{bmatrix}^T . \tag{7e}$$

The new input e_c is the stepoint of the elevation. The continuous model is then converted to a time discrete model

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u} \tag{8}$$

with the forward Euler method

$$\mathbf{A} = \mathbf{I}_{6\times 6} + \mathbf{A}_c\Delta t \tag{9a}$$
$$\mathbf{B} = \mathbf{B}_c\Delta t \tag{9b}$$

with Δt being the sampling time.

The cost function

$$\phi = \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + q_1 p_{ci}^2 + q_2 e_{ci}^2 \tag{10}$$

is used as a minimization criteria, with the final value for the travel $\lambda_f = 0$ and $q_1 = 1$ and $q_2 = 2$. The values for q_1 and q_2 are chosen this way to

reduce the oscillations in the opimal trajectory of p and $\dot{p}$ which occur if $q_1 = q_2 = 1$ is used.

The initial value $\mathbf{x}_0 = \begin{bmatrix} \pi & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ is used to ensure a travel distance of π.

As in section 4 a constraint of $\pm 30°$ is used for the pitch p and the setpoint of the pitch p_c. Input constraints of $\pm 60°$ for e_c are added to avoid a collision between the helicopter and the table on which the helicopter is mounted. Since (9) needs to be valid at each time step the equaions are added as equality constraints

$$\mathbf{A}_{eq} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & -\mathbf{B} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ -\mathbf{A} & \mathbf{I} & \ddots & & \vdots & \mathbf{0} & \ddots & \ddots & & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{0} & \vdots & & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & -\mathbf{A} & \mathbf{I} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & -\mathbf{B} \\ \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{I} & \mathbf{0} & \cdots & \cdots & \cdots & \mathbf{0} \end{bmatrix} \tag{11a}$$

$$\mathbf{B}_{eq} = \begin{bmatrix} \mathbf{A}\mathbf{x}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{x}_f \end{bmatrix}, \tag{11b}$$

with $\mathbf{x}_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ being the final state at $t = N$. A nonlinear constraint

$$c(\mathbf{x}_k) = \alpha \exp\left(-\beta(\lambda_k - \lambda_t)^2\right) - e_k \leq 0 \quad \forall k \in \{1, \ldots, N\} \tag{12}$$

with $\alpha = 0.2$, $\beta = 20$ and $\lambda_t = \frac{2\pi}{3}$, is added. Since (12) is nonlinear the optimization problem is nonlinear and therefore a nonlinear solver is used. The MATLAB command `fmincon` with three different algorithms is used. The SQP algorithm converges to a solution where the tarjectory of the travel λ consists of a single step from π to 0, which is unphysical and therefore cannot be used as a reference trajectory for the helicopter. The active-set method converges to a solution where the input u_1 is 83 % of the time at saturation limit. Although this is only the open loop trajectory it means that when the loop is closed using the LQR the input is still at the saturation limit (assuming that the model is perfect and that there are no disturbances) which means that the control loop is actually open. Because of that the interior-point method is used which results in an trajectory where the input is only 8 % of the time at the saturation limit. The computation time for calculating the trajectory are 0.33 s for the SQP method, 8.06 s for the active-set method and 53.51 s for the interior-point method, but this is

25

of little concern since the optimization problem doesn't need to be solved online.

The code for calculating the optimal trajectory is shown in appendix A.3, the nonlinear constraint function can be seen in appendix A.4 and the calculation of the LQR gain matrix is done in appendix A.5. The simulink diagram is shown in fig. 24 and fig. 25.

The time curve of the helicopter without feedback can be seen in fig. 18. As in section 4 the trajectory of the pitch p is followed, which is due to the pitch control loop which helps to counteract for modeling errors and that a linear model of the nonlinear system is used. The same applies for the elevation control loop which had a reference point of zero in the last two sections and has a trajectory unequal to zero due to the nonlinear constraint (12) in this section. The reference trajectory of the travel λ is not followed satisfactorily. This is the case because the travel λ is constrolled in open loop and due to modeling errors the input is not calculated correctly which causes the severe deviations.

As in section 5 an LQR is used as feedback controller. The input is then calculated by

$$\mathbf{u}_t = \mathbf{u}_t^* - \mathbf{K}\left(\mathbf{x}_t - \mathbf{x}_t^*\right) \tag{13}$$

with $\mathbf{u}_t^*$ being the optimal input sequence and $\mathbf{x}_t^*$ being the optimal trajectory of the states. This new input sequence $\mathbf{u}$ is then used instead of the optimal input trajectory $\mathbf{u}_*$ to ensure that the deviations of the desired travel trajectory λ is reduced. The weighting matrices

$$\mathbf{Q} = \mathrm{diag}\left(5, 1, 1, 1, 1, 1\right) \tag{14a}$$

$$\mathbf{R} = \mathrm{diag}\left(1, 1\right) \tag{14b}$$

are used, which results in a feedback matrix

$$\mathbf{K} = \begin{bmatrix} -0.1899 & -0.6725 & -0.7316 & 0.0272 & -0.0000 & -0.0000 \\ 0.0000 & 0.0000 & -0.0000 & -0.0000 & 0.0892 & 0.4678 \end{bmatrix} . \tag{15}$$

The values for $\mathbf{Q}$ and $\mathbf{R}$ are chosen such that the trajectory of the travel λ is followed with small deviations and that there are no oscillations. Higher weights on the travel λ cause oscillations. Higher weights on the pitch p cause large deviations in the travel, since the controller tries to decrease deviations between the pitch p and the optimal pitch trajectory p^*. Higher weights on the input u_1 has approximately the same effect as higher weights on the pitch p. The weight on the input u_2 influences the behaviour only minimally.

The time curve of the helicopter with an LQR as feedback controller is shown in fig. 19. The deviations from the trajectory of the travel λ are much smaller than compared to the ones in fig. 18. Apart from that there are less oscillations in the time curve of the elevation e.
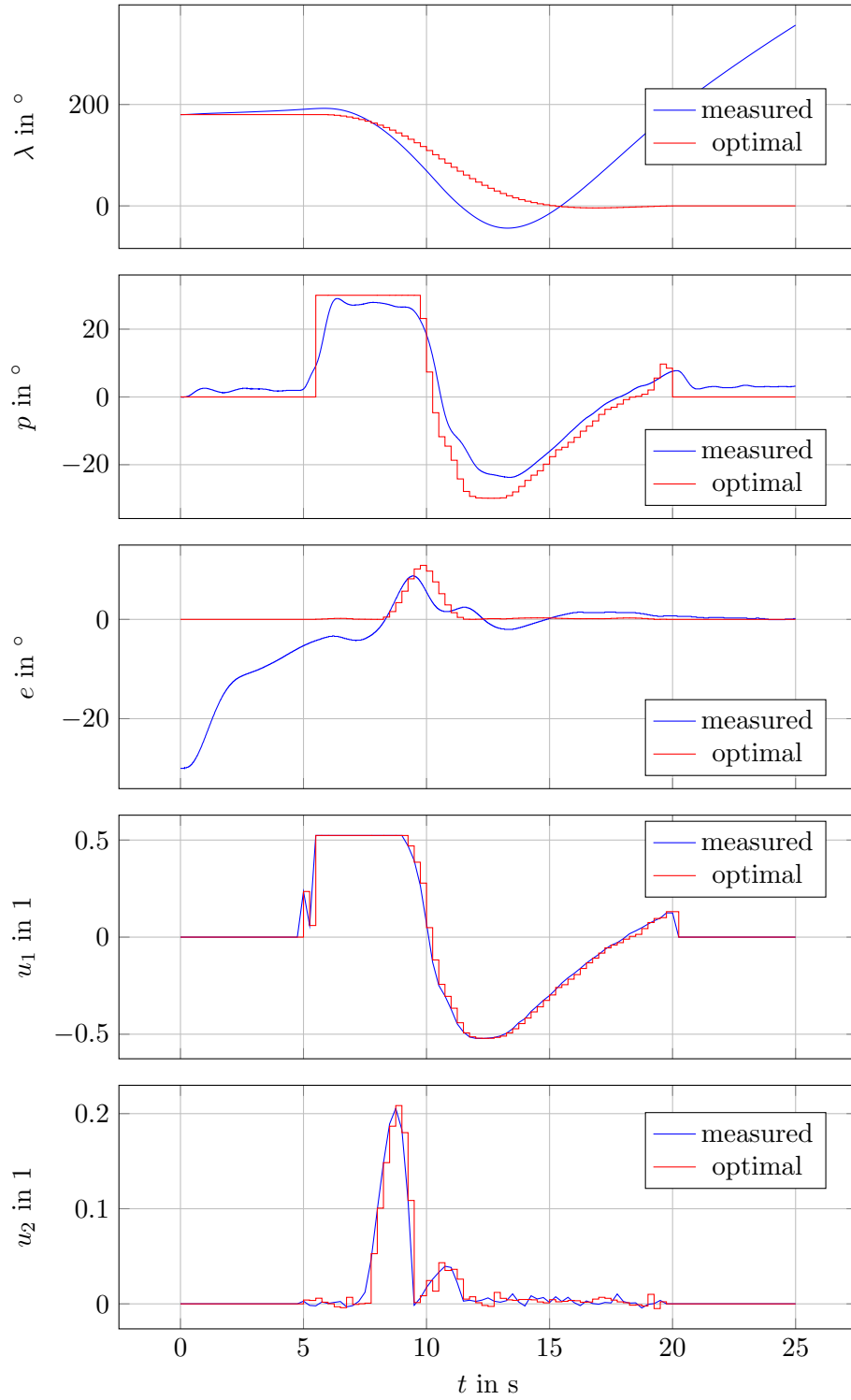
Figure 18: Time curve of inputs and states without using feedback.

The constraint on the elevation is not satsfied perfectly due to the coupling of the pitch p and travel λ with the elevation e which is not considered in the simplified model which is used in this laboratory. The optimal input sequences don't incorporate the coupling because of the simplified model and thereby cause the deviations. The decoupling of the simplified model can also be observed in the matrix $\mathbf{K}$ which has a block diagonal structure. The coupling of the two subsystems can be observed when e.g. the pitch is 90°, both turbines have no effect on the elevation since the direction of force and the direction of the elevation angle are orthogonal. A way to improve the performance would be to use a better model for the optimization problem which incorporates the coupling.

The deviations from time curve of the pitch p are larger but they are negligible since the goal is to control the travel λ and to satisfy the constraints. Due to the feedback the noise of the measurement can be seen in in the input trajectory. The impulses that can be observed in the input u_2 occur because the rates r, $\dot{p}$ and $\dot{e}$ are estimated using filters of the type

$$G\left(s\right) = \frac{Ts}{s + T} \tag{16}$$

which have a differentiating behaviour for frequencies lower than $\omega = \frac{1}{T}$. Since the measurements of λ, p and e have discrete values a step occurs whenever the value changes which results in an impulse in the velocity estimation.

To improve the optimal trajectory constraints on r and $\dot{e}$ are introduced. The constraints are

$$-0.15 \leq r \leq 0.15 \tag{17a}$$

$$-0.12 \leq e \leq 0.12 \; . \tag{17b}$$

The time curve with additional constraints on r and $\dot{e}$ is shown in fig. 20. The number of steps N is increased to 100 to get a feasible solution. The reason behind this is that there needs to be a certain speed r if the objective is to move from π to 0 within N steps. The increased number of steps cause the computation time to rise to $2100\,\mathrm{s}$ compared to $50\,\mathrm{s}$ which are needed for computation of the trajectory in fig. 18 and fig. 19. The deviations of the optimal travel λ trajectory are smaller than in fig. 19, although some oscillation occur which are caused by a too large gain in the $\mathbf{K}$ matrix, so a smaller weight on
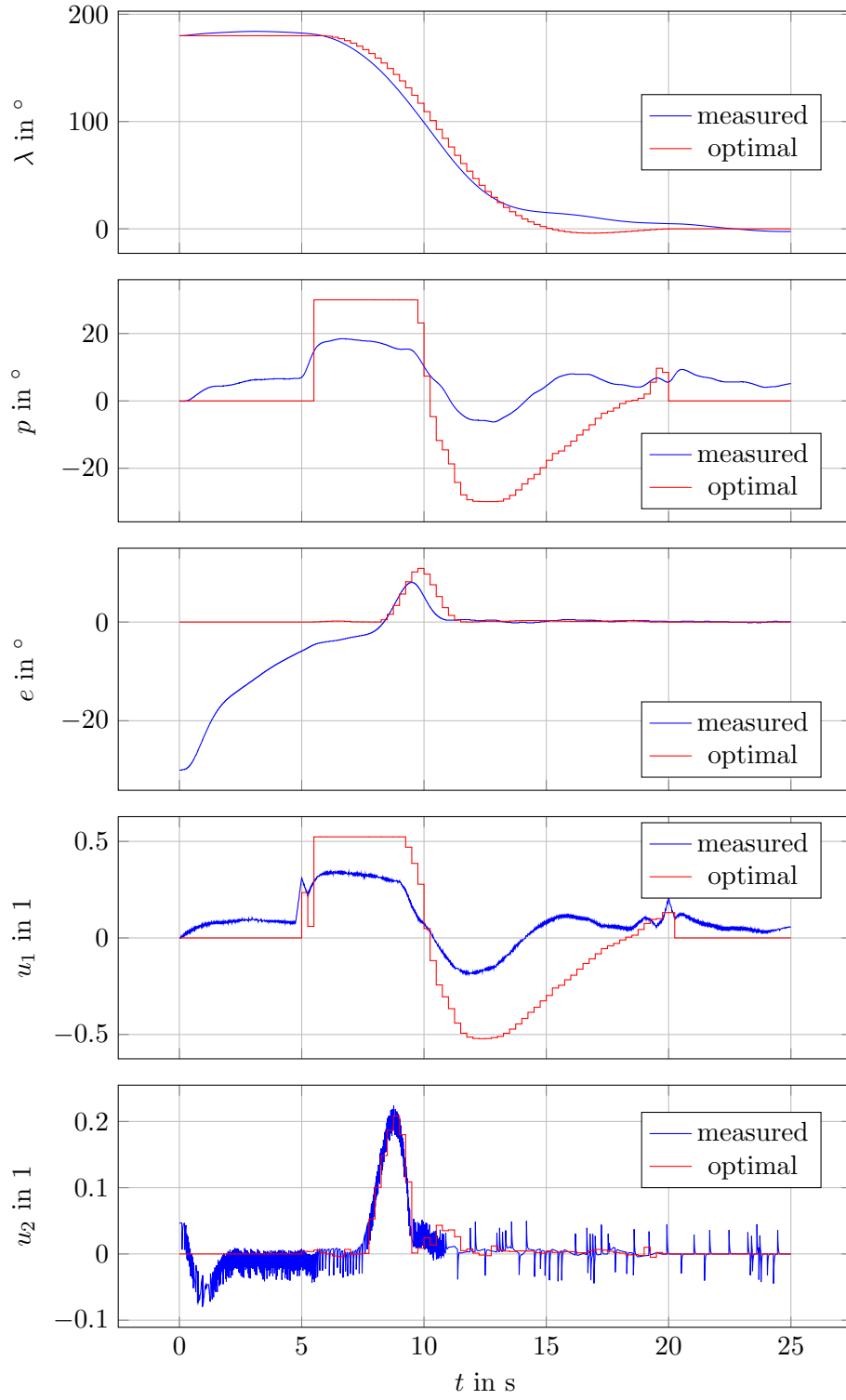
Figure 19: Time curve of inputs and states while using (15) as feedback controller.
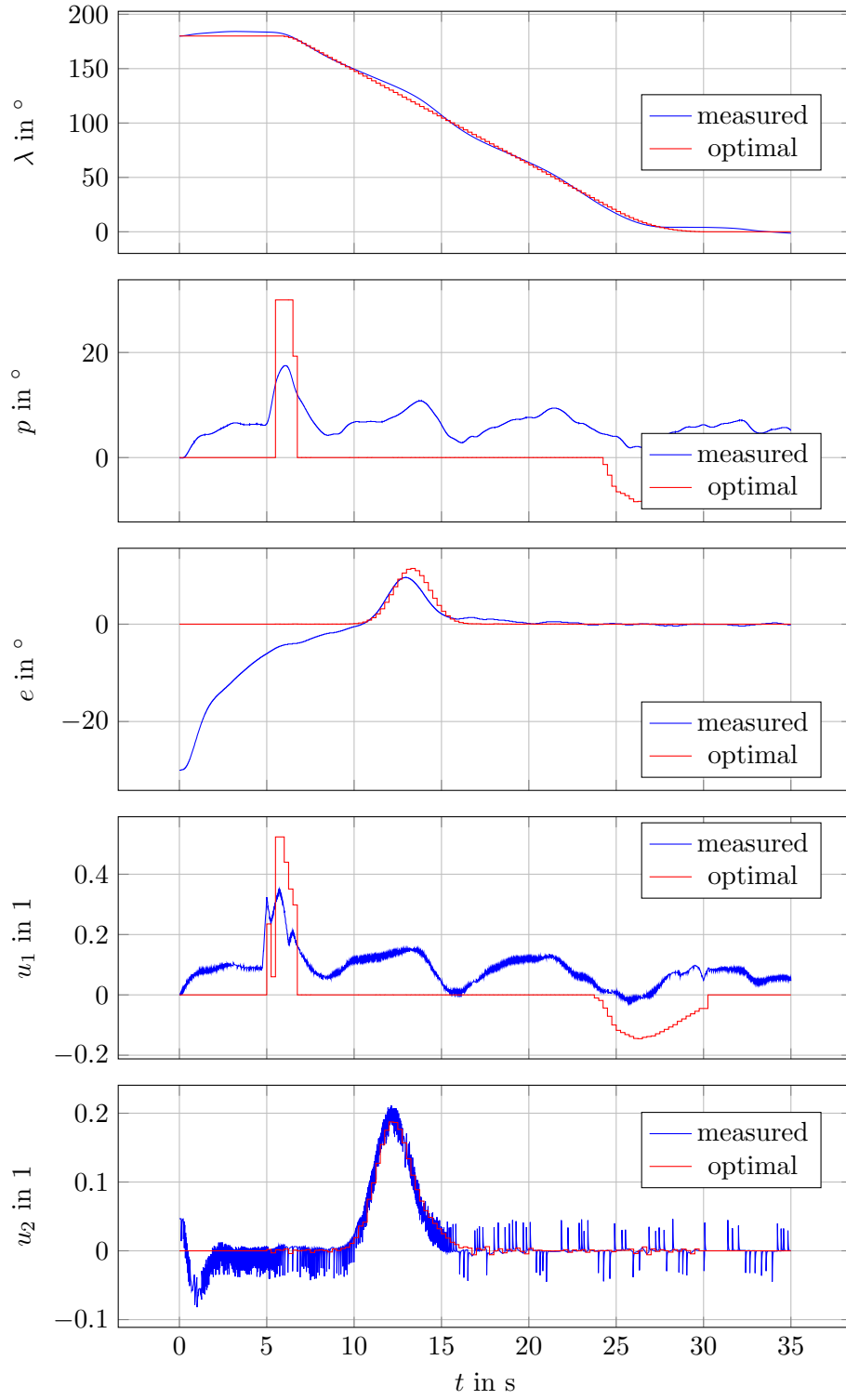
Figure 20: Time curve of inputs and states while using the optimal trajectory with additional constraints (17).

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

Using the modeled helicopter at the lab. We were able to try to control the helicopter with the different approaches stated from before.

When using only an optimal input sequence with no feedback we would get quite an unstable system as seen in the result above. The reason why this did not work so well is because the model we use for the helicopter is quite simplified and does not reflect all the coupling and dynamics of the system being controlled.

However when we used an optimal input sequence combined with control theory using an LQR in a feedback loop our results drastically improved our results. Also this solution did not require a long computation time as, but sacrifices the ability to impose hard constraints on input and states as the LQR does not limit its output.

Therfor if we need constraints on the feedback a solution would be to use an MPC. However it would be a lot more computational intensive and would require better/faster or specialized hardware to work seamlessly.

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

A MATLAB Code

A.1 problem2.m

```matlab
% *****************************************************
% *                                                   *
% *         Optimization and Control                  *
% *                                                   *
% *         Helikopterlab                             *
% *                                                   *
% * problem2.m                                        *
% *                                                   *
% *                                                   *
% * Updated on 04/2009 by Agus Ismail Hasan           *
% *                                                   *
% *****************************************************

init01;
delta_t  = 0.25;                        % sampling time
sek_forst = 5;

% System model. x=[lambda r p p_dot]'

Ac = [0 1 0 0;
      0 0 -K_2 0;
      0 0 0 1;
      0 0 -K_1*K_pp -K_1*K_pd];
Bc = [0; 0; 0; K_1*K_pp];

A1 = eye(4) + Ac*delta_t;
B1 = Bc*delta_t;

% Number of states and inputs

mx = size(A1,2);  % Number of states
mu = size(B1,2);  % Number of inputs

% Initial values

x1_0 = pi;                              % Lambda
x2_0 = 0;                               % r
x3_0 = 0;                               % p
x4_0 = 0;                               % p_dot
x0   = [x1_0 x2_0 x3_0 x4_0]';          % Initial values
```

```matlab
41  xf    = [0 0 0 0]';
42
43  % Time horizon and initialization
44
45  N  = 100;                     % Time horizon for states
46  M  = N;                       % Time horizon for inputs
47  z  = zeros(N*mx+M*mu,1);% Initialize z for the whole horizon
48  z0 = z;                 % Initial value for optimization
49
50  % Bounds
51
52  ul      = -30*pi/180;    % Lower bound on control -- u1
53  uu      = 30*pi/180;     % Upper bound on control -- u1
54
55  xl      = -Inf*ones(mx,1); % Lower bound on states
56  xu      = Inf*ones(mx,1);  % Upper bound on states
57  xl(3)   = ul;              % Lower bound on state x3
58  xu(3)   = uu;              % Upper bound on state x3
59
60  % Generate constraints on measurements and inputs
61
62  vlb       = [kron(ones(N,1),xl);kron(ones(N*mu,1),ul)];
63  vub       = [kron(ones(N,1),xu);kron(ones(N*mu,1),uu)];
64  vlb(N*mx+M*mu)  = 0;
65  vub(N*mx+M*mu)  = 0;
66
67  % Generate the matrix Q and the vector c
68
69  Q1 = zeros(mx,mx);
70  Q1(1,1) = 1;                  % Weight on state x1
71  %Q1(2,2) = ;                  % Weight on state x2
72  Q1(3,3) = 0;                  % Weight on state x3
73  %Q1(4,4) = ;                  % Weight on state x4
74  P1 = 10;                      % Weight on input
75  Q = 2*genq2(Q1,P1,N,M,mu);    % Generate Q
76  c = zeros(N*mx+M*mu,1);       % Generate c
77
78  % Generate system matrixes for linear model
79  Aeq = gena2(A1,B1,N,mx,mu);
80  Aeq = [ Aeq; [zeros(mx,(N-1)*mx), eye(4), zeros(mx,N*mu)]];
81
82  beq = [A1*x0; zeros((N-1)*mx,1);xf];      % Generate b
83
84  % Solve Qp problem with linear model
```

```matlab
85  tic
86  [z,lambda] = quadprog(Q,[],[],[],Aeq,beq,vlb,vub);
87  t1=toc;
88
89  % Calculate objective value
90
91  phi1 = 0.0;
92  PhiOut = zeros(N*mx+M*mu,1);
93  for i=1:N*mx+M*mu
94    phi1=phi1+Q(i,i)*z(i)*z(i);
95    PhiOut(i) = phi1;
96  end
97
98  % Extract control inputs and states
99
100 u  = [z(N*mx+1:N*mx+M*mu);z(N*mx+M*mu)];
101
102 x1 = [x0(1);z(1:mx:N*mx)];    % State x1 from solution
103 x2 = [x0(2);z(2:mx:N*mx)];    % State x2 from solution
104 x3 = [x0(3);z(3:mx:N*mx)];    % State x3 from solution
105 x4 = [x0(4);z(4:mx:N*mx)];    % State x4 from solution
106
107 Antall = 5/delta_t;
108 Nuller = zeros(Antall,1);
109 Enere  = ones(Antall,1);
110
111 u  = [Nuller; u; Nuller];
112 x1 = [pi*Enere; x1; Nuller];
113 x2 = [Nuller; x2; Nuller];
114 x3 = [Nuller; x3; Nuller];
115 x4 = [Nuller; x4; Nuller];
116
117 %save trajektor1ny
118
119 % figure
120 t = 0:delta_t:delta_t*(length(u)-1);    % real time
121
122 figure(2)
123 subplot(511)
124 stairs(t,u),grid
125 ylabel('u')
126 subplot(512)
127 plot(t,x1,'m',t,x1,'mo'),grid
128 ylabel('lambda')
```

```
129  subplot(513)
130  plot(t,x2,'m',t,x2','mo'),grid
131  ylabel('r')
132  subplot(514)
133  plot(t,x3,'m',t,x3','mo'),grid
134  ylabel('p')
135  subplot(515)
136  plot(t,x4,'m',t,x4','mo'),grid
137  xlabel('tid (s)'),ylabel('pdot')
```

A.2 problem3.m

```
1  %% Optimal control of Pitch/Travel with Feedback(LQ)
2  problem2;
3
4  % [lambda r p p_dot]
5  LQR.Q = diag([1, 0, 10, 0]);
6  LQR.R = 0.5;
7
8  K = dlqr(A1, B1, LQR.Q, LQR.R);
```

A.3 problem4.m

```
1  % ****************************************************
2  % *                                                  *
3  % *         Optimization and Control                 *
4  % *                                                  *
5  % *         Helikopterlab                            *
6  % *                                                  *
7  % * problem4.m                                       *
8  % *                                                  *
9  % *                                                  *
10 % * Updated on 04/2009 by Agus Ismail Hasan          *
11 % *                                                  *
12 % ****************************************************
13
14 init01;
15 delta_t   = 0.25;                    % sampling time
16 sek_forst = 5;
17
18 % System model. x=[lambda r p p_dot e e_dot]'
19 % u = [p_c, e_c]
20
21 Ac = [ 0 1 0 0 0 0;
22        0 0 -K_2 0 0 0;
```

```matlab
23          0 0 0 1 0 0;
24          0 0 -K_1*K_pp -K_1*K_pd 0 0
25          0 0 0 0 0 1;
26          0 0 0 0 -K_3*K_ep, -K_3*K_ed];
27  Bc = [0 0;
28          0 0 ;
29          0 0 ;
30          K_1*K_pp 0;
31          0 0;
32          0 K_3*K_ep];
33
34  A1 = eye(6) + Ac*delta_t;
35  B1 = Bc*delta_t;
36
37  % Number of states and inputs
38
39  mx = size(A1,2);  % Number of states
40  mu = size(B1,2);  % Number of inputs
41
42  % Initial values
43
44  x1_0 = pi;                          % Lambda
45  x2_0 = 0;                           % r
46  x3_0 = 0;                           % p
47  x4_0 = 0;                           % p_dot
48  x5_0 = 0;                           % e
49  x6_0 = 0;                           % e_dot
50  x0 = [x1_0 x2_0 x3_0 x4_0 x5_0 x6_0]'; %Initial values
51  xf = [0 0 0 0 0 0]';% final values
52
53  % Time horizon and initialization
54
55  N  = 60;          % Time horizon for states
56  M  = N;           % Time horizon for inputs
57  z  = [x0; kron(ones(N-1,1), [0;0;0;0;0;0.1]); ...
58      kron(ones(N,1), [0;0.1])];
59  z0 = z;           % Initial value for optimization
60
61  % Bounds
62
63  ul      = [-30*pi/180; -60*pi/180]; % Lower bound on u
64  uu      = [30*pi/180; 60*pi/180];   % Upper bound on u
65
66  xl      = -Inf*ones(mx,1); % Lower bound on states
```

```matlab
67  xu      = Inf*ones(mx,1);   % Upper bound on states
68  xl(3)   = ul(1);            % Lower bound on state x3
69  xu(3)   = uu(1);            % Upper bound on state x3
70  %xl(2)   = -0.15;
71  %xu(2)   = 0.15;
72  %xl(6)   = -0.12;
73  %xu(6)   = 0.12;
74
75  % Generate constraints on measurements and inputs
76
77  vlb           = [kron(ones(N,1),xl);kron(ones(N,1),ul)];
78  vub           = [kron(ones(N,1),xu);kron(ones(N,1),uu)];
79  vlb(N*mx+M*mu) = 0; %We want the last input to be zero
80  vub(N*mx+M*mu) = 0; %We want the last input to be zero
81
82
83  % Generate the matrix Q and the vector c
84
85  Q1 = zeros(mx,mx);
86  Q1(1,1) = 1;                     % Weight on state x1
87  %Q1(2,2) = ;                     % Weight on state x2
88  Q1(3,3) = 0;                     % Weight on state x3
89  %Q1(4,4) = ;                     % Weight on state x4
90  P1 = diag([1,2]);                % Weight on input
91  Q = genq2(Q1,P1,N,M,mu);         % Generate Q
92  c = zeros(N*mx+M*mu,1);          % Generate c
93
94  % Generate system matrixes for linear model
95  Aeq = gena2(A1,B1,N,mx,mu);
96  Aeq = [ Aeq; [zeros(mx,(N-1)*mx), eye(6), ...
97      zeros(mx,N*mu)]];
98
99  beq = [A1*x0; zeros((N-1)*mx,1); xf];    % Generate b
100
101 % Solve nonlinear problem with linear model
102 phi = @ (x) (x'*Q*x);
103 options = optimset('Display','notify', ...
104     'Diagnostics','on',...
105     'MaxFunEvals',Inf,'MaxIter',Inf);
106 tic
107 [z, lambda] = fmincon(phi, z0,[],[],Aeq,beq,vlb,...
108     vub,@constr4,options);
109 t1=toc
110
```

```matlab
111
112  % Calculate objective value
113
114  phi1 = 0.0;
115  PhiOut = zeros(N*mx+M*mu,1);
116  for i=1:N*mx+M*mu
117    phi1=phi1+Q(i,i)*z(i)*z(i);
118    PhiOut(i) = phi1;
119  end
120
121  % Extract control inputs and states
122
123  u1 = [z(N*mx+1:mu:N*mx+M*mu);z(N*mx+M*mu-1)];
124  u2 = [z(N*mx+2:mu:N*mx+M*mu);z(N*mx+M*mu)];
125
126  x1 = [x0(1);z(1:mx:N*mx)];      % State x1 from solution
127  x2 = [x0(2);z(2:mx:N*mx)];      % State x2 from solution
128  x3 = [x0(3);z(3:mx:N*mx)];      % State x3 from solution
129  x4 = [x0(4);z(4:mx:N*mx)];      % State x4 from solution
130  x5 = [x0(4);z(5:mx:N*mx)];      % State x5 from solution
131  x6 = [x0(4);z(6:mx:N*mx)];      % State x6 from solution
132
133  Antall = 5/delta_t;
134  Nuller = zeros(Antall,1);
135  Enere  = ones(Antall,1);
136
137  u1  = [Nuller; u1; Nuller];
138  u2  = [Nuller; u2; Nuller];
139  x1  = [pi*Enere; x1; Nuller];
140  x2  = [Nuller; x2; Nuller];
141  x3  = [Nuller; x3; Nuller];
142  x4  = [Nuller; x4; Nuller];
143  x5  = [Nuller; x5; Nuller];
144  x6  = [Nuller; x6; Nuller];
145  u =  [u1 u2];
146  %save trajektor1ny
147  %%
148   K = 0;
149  % figure
150  t = 0:delta_t:delta_t*(length(u1)-1);      % real time
151
152
153  figure(2)
154  subplot(511)
```

39

```matlab
155  stairs(t,u1),grid
156  ylabel('u1')
157  subplot(512)
158  plot(t,x1,'m',t,x1,'mo'),grid
159  ylabel('lambda')
160  subplot(513)
161  plot(t,x2,'m',t,x2,'mo'),grid
162  ylabel('r')
163  subplot(514)
164  plot(t,x3,'m',t,x3,'mo'),grid
165  ylabel('p')
166  subplot(515)
167  plot(t,x4,'m',t,x4,'mo'),grid
168  xlabel('tid (s)'),ylabel('pdot')
169
170  figure(3)
171  subplot(311)
172  stairs(t,u2),grid
173  ylabel('u2')
174  subplot(312)
175  plot(t,x5,'m',t,x5,'mo'),grid
176  ylabel('e')
177  subplot(313)
178  plot(t,x6,'m',t,x6,'mo'),grid
179  ylabel('e_dot')
```

A.4   constr4.m

```matlab
1  function [ c, ceq ] = constr4( x )
2
3      N = 60;
4      mx = 6;
5      alpha = 0.2;
6      beta = 20;
7      lambda_t=2*pi/3;
8
9      lambda_k = x(1:6:N*mx);
10     e_k = x(5:6:N*mx);
11
12     c = alpha*exp(-beta*((lambda_k-lambda_t).^2))-e_k;
13     ceq = [];
14  end
```

A.5   problem4_qr.m

```matlab
% Optimal control of Pitch/Travel and Elevation
% with Feedback(LQ)

problem4;

%%

LQR.Q = diag([5, 1, 1, 1, 1, 1]);
LQR.R = diag([1 1]);

K = dlqr(A1, B1, LQR.Q, LQR.R);
```

B   Simulink Diagrams
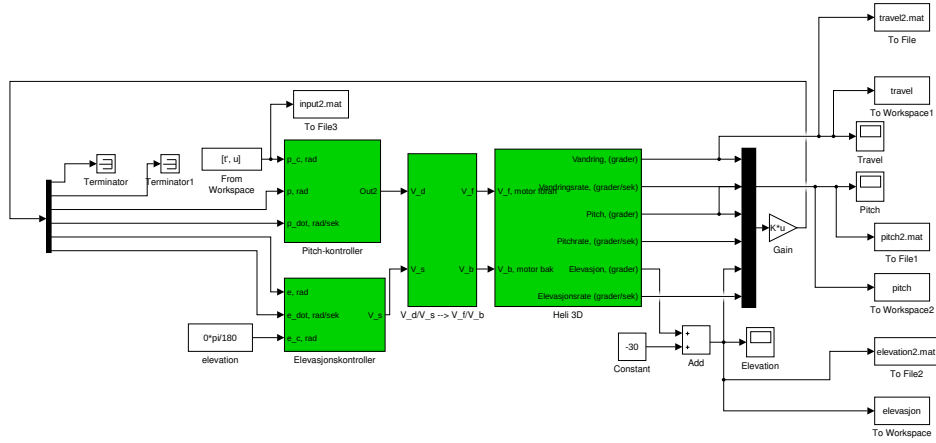
B.1   Problem 2

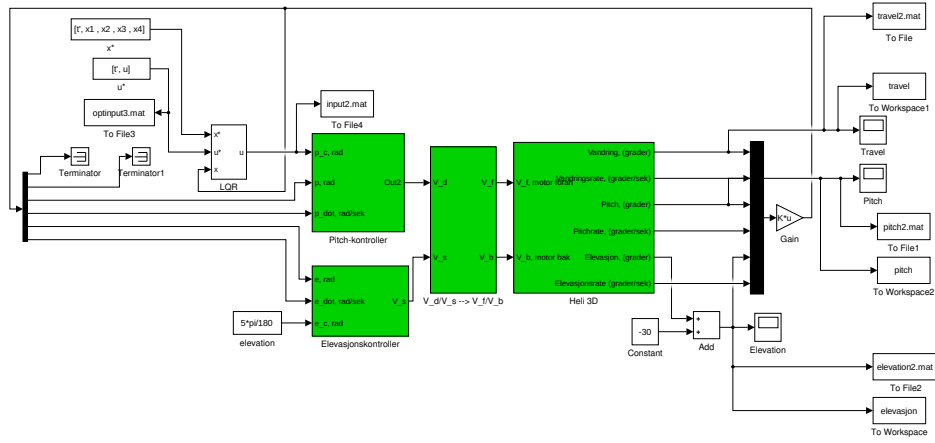Figure 21: Simulink diagram of problem 2.

B.2   Problem 3

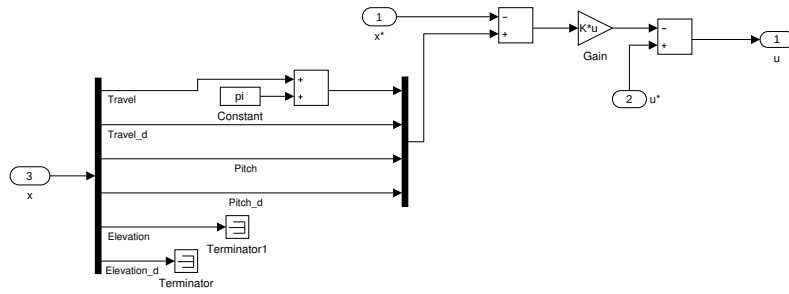

Figure 22: Simulink diagram of problem 3.

Figure 23: Simulink diagram of LQR subsystem of problem 3.
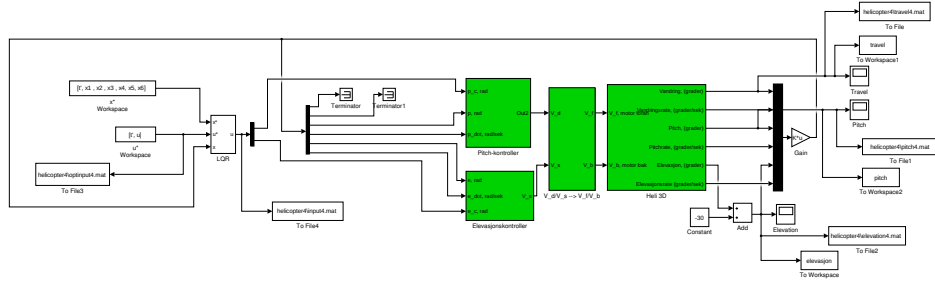
43

B.3   Problem 4
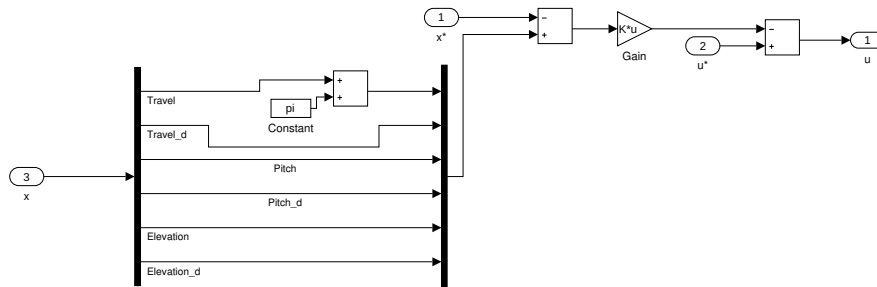

Figure 24: Simulink diagram of problem 4.

Figure 25: Simulink diagram of LQR subsystem of problem 4.

Bibliography

Berland, H. (2010). *En introduksjon til Latex*. `http://www.pvv.ntnu.no/~berland/latex/latexintro4up.pdf`.

Downes, M. (2002). *Short Math Guide for LATEX*. `ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`.

Oetiker, T., Partl, H., Hyna, I., and Schlegl, E. (2011). *The Not So Short Introduction to LATEX 2e*. `http://tobi.oetiker.ch/lshort/lshort.pdf`.