TTK4135 Optimization and Control
Lab Report

716120, 723987
Group 1

April 21, 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

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

Discussing the lab setup: Dual rotor movable arm, differential thrust, etc.

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

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

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

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

Note simplifications and limitations:

- Travel rate proportional to pitch, otherwize decoupled
- Pitch has no counteracting 'spring' force

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, i.e: see problem2...

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	$\mathrm{kg\,m^2}$
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

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

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...

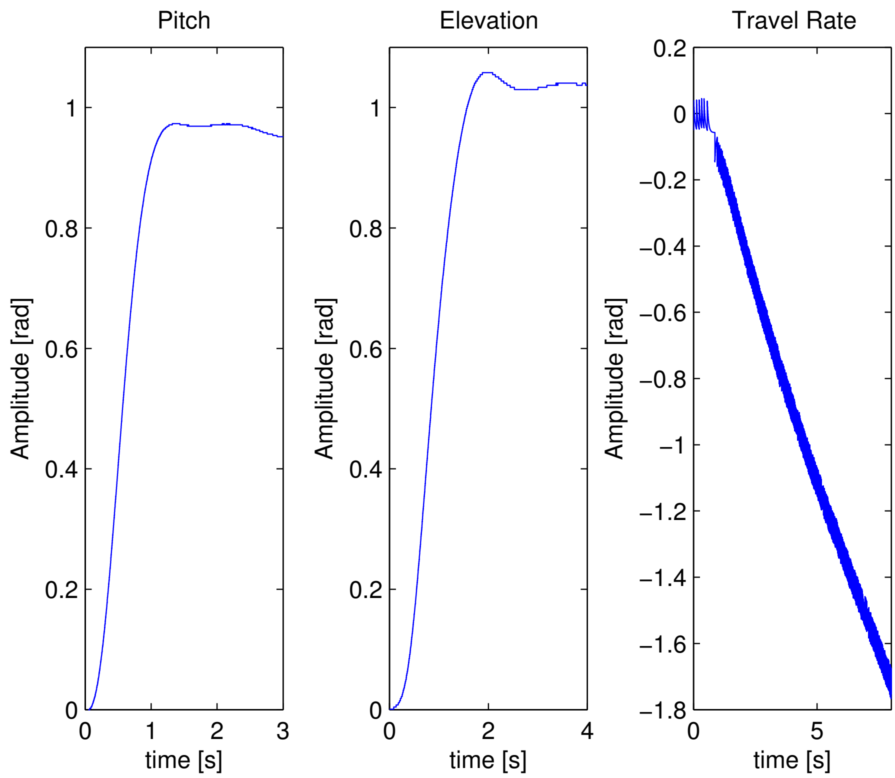

Figure 1: Step response.

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

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -K_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -K_1K_{pp} & -K_1K_{pd} \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ K_1K_{pp} \end{bmatrix} p_c$$

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...

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

Discuss differences with textbook model

4.2 Discretization

Let $x = \begin{bmatrix} \lambda & \dot{\lambda} & p & \dot{p} \end{bmatrix}^\top$, $u = p_c$, and A_c, B_c denote the state transition matrix and the **(Whatever B is called)** from (3) respectively. Then, discretizing (3) using forward Euler with a time step Δt we get the discrete time state space model

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \dot{x}_k && \text{(4a)} \\ &= x_k + \Delta t(A_c x_k + B_c u_k) && \text{(4b)} \\ &= (I + \Delta t A_c)x_k + (\Delta t B_c)u_k && \text{(4c)} \\ &= Ax_k + Bu_k && \text{(4d)} \end{aligned}$$

where $x_k = x(k\Delta t)$ and $u_k = u(k\Delta t)$.

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

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

for a chosen q penalizing input expendature in favour of minimizing deviations in the travel trajectory, and vice verca for $q < 1$. (Copied prev eq from assignment, but are the indexes correct?) Equivalently we can define (5) in terms of the full state and input variables.

$$\phi = \frac{1}{2} \sum_{i=0}^{N-1} (x_{i+1} - x_f)^\top Q (x_{i+1} - x_f) + u_i^\top R u_i, \tag{6}$$

where

$$Q = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ R = 2q.$$

We need to choose different symbols than Q,R, but I don't know which...

The system dynamics (4) subjects (6) to the linear equality constraints

$$\underbrace{\left[\begin{array}{cccc|cccc} I & & & & -B & & & \\ -A & \ddots & & & & \ddots & & \\ & \ddots & \ddots & & & & \ddots & \\ & & -A & I & & & & -B \end{array}\right]}_{A_eq} \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} -Ax_f \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B_eq}. \tag{7}$$

We let $z = \begin{bmatrix} x_1 & \dots & x_N & u_0 & \dots & u_{N-1} \end{bmatrix}^\top$ and define

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

We impose lower bounds $p^{\text{low}} = -\frac{30\pi}{180}$ and upper bounds $p^{\text{high}} = \frac{30\pi}{180}$ on the pitch state and set-point, and state the resulting QP-problem in terms of the optimization variable z:

$$\min_z \quad \frac{1}{2} z^\top G z \tag{8a}$$

subject to

$$A_{eq}z = B_{eq}, \tag{8b}$$

$$p^{\text{low}} \leq p_k \leq p^{\text{high}}, \quad k \in \{1, \ldots, N\}. \tag{8c}$$

(8) is solved using MATLAB's `quadprog`. 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^*.

4.4 Results and discussion

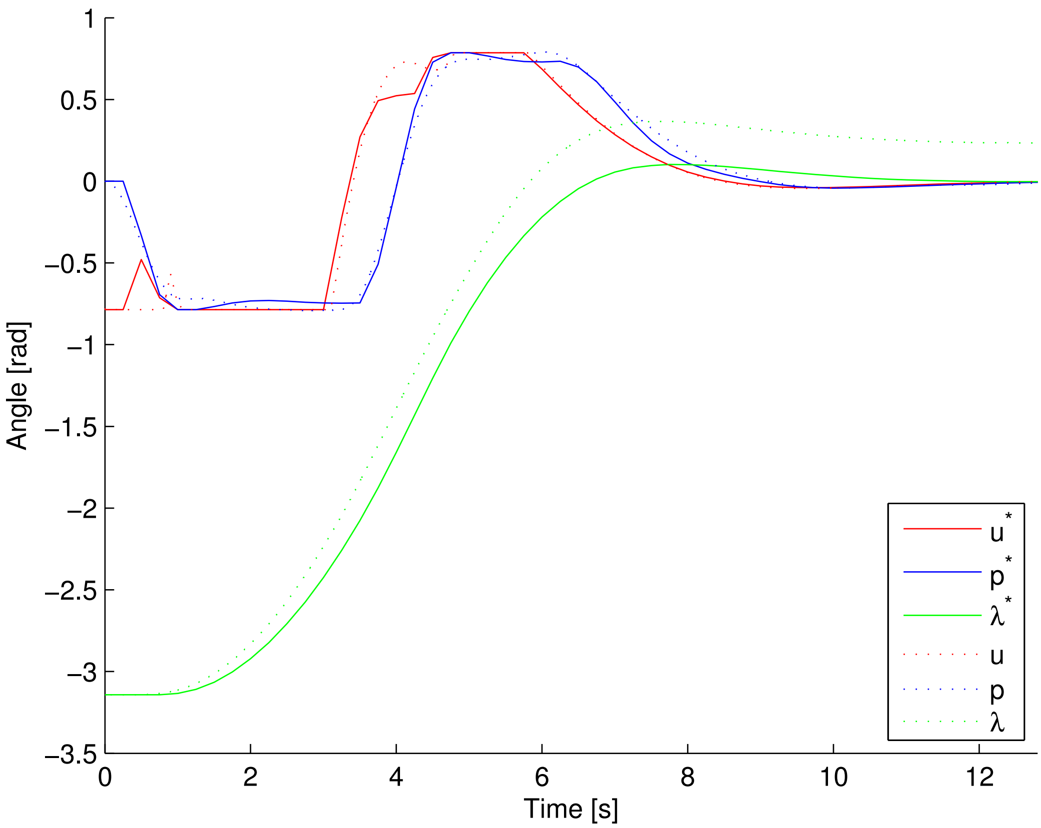

Figure 2: Optimal vs. measured trajectory and input sequence.

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:

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

or, alternatively

$$\Delta u_k = -K^\top \Delta x_k, \tag{9}$$

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 (9) is the optimal solution minimizing the quadratic objective function

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

subject to the system dynamics (4), where

$$K = (R + B^\top P B)^{-1} B^\top P A, \tag{10}$$

and P is the unique positive definite solution to the discrete time algebraic Riccati equation.(10) 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.

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

5.2 Results and discussion

5.2.1 MPC discussion

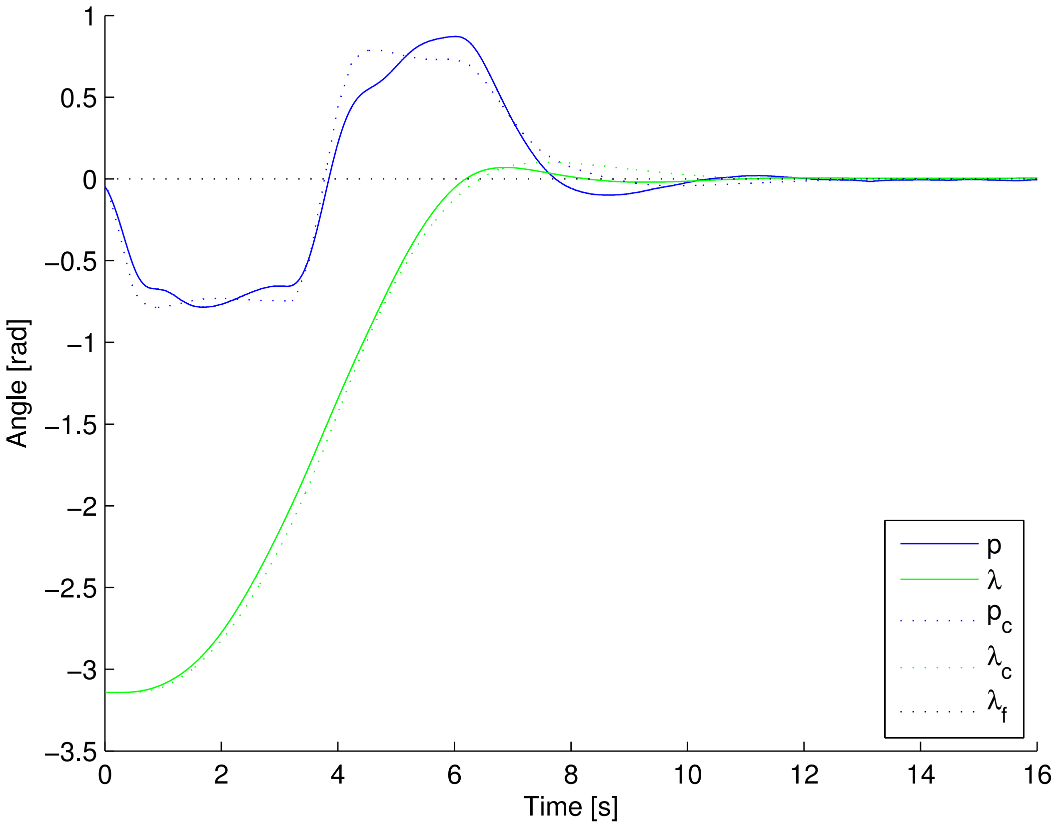

Figure 3: LQR

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

$$\begin{bmatrix} \dot{\lambda} \\ \ddot{\lambda} \\ \dot{p} \\ \ddot{p} \\ \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.03 & -0.39 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -7.13 & -3.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -3.03 & -2.44 \end{bmatrix} \begin{bmatrix} \lambda \\ \dot{\lambda} \\ p \\ \dot{p} \\ e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6.74 & 0 \\ 0 & 0 \\ 0 & 3.13 \end{bmatrix} \begin{bmatrix} p_c \\ e_c \end{bmatrix} \tag{11}$$

Discuss decoupling...?

6.2 Discretization

Now let $x = \begin{bmatrix} \lambda & \dot{\lambda} & p & \dot{p} & e & \dot{e} \end{bmatrix}^\top$, $u = \begin{bmatrix} p_c & e_c \end{bmatrix}^\top$, and A_c, B_c denote the state transition matrix and ** matrix in (11), respectively. Again, using approximate discretization via Euler we obtain a discrete state space model

$$x_{k+1} = (I + \Delta t A_c)x_k + (\Delta t B_c)u_k. \tag{12}$$

6.3 Optimization problem with nonlinear constraints

We wish to 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

$$\phi = \sum_{i=1}^{N} (\lambda_i - \lambda_f)^2 + q_1 p_{c_i}^2 + q_2 e_{c_i}^2, \quad q_1, q_2 \geq 0,$$

or alternatively

$$\phi = \sum_{i=0}^{N-1} (x_{i+1} - x_f)^\top Q (x_{i+1} - x_f) + u_i^\top R u_i, \tag{13}$$

where

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; R = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}. \tag{14}$$

The objective function (13) is subject to the system dynamics (12) and we impose linear equality constraints similarily to (7). Additionally we impose a non-linear inequality constraint on the elevation for every time step:

$$e_k \geq \alpha \exp\left(-\beta\left(\lambda_k - \lambda_t\right)^2\right), \quad k = \{1, \ldots, N\}, \tag{15}$$

where we let $\alpha = 0.2$, $\beta = 20$. Because of the non-linearity of (15) we can no longer use a QP-solver, and the following optimization problem in terms of the optimization variable $z = \begin{bmatrix} x_1 & \ldots & x_N & u_0 & \ldots & u_{N-1} \end{bmatrix}$ is solved using MATLAB's `fmincon`

$$\min_{z} \quad z^T G z \tag{16a}$$

subject to

$$A_{eq} z = B_{eq}, \tag{16b}$$

$$\alpha \exp\left(-\beta\left(\lambda_k - \lambda_t\right)^2\right) - e_k \leq 0, \tag{16c}$$

$$-\frac{25\pi}{180} \leq p_k \leq \frac{25\pi}{180} \tag{16d}$$

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.

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

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

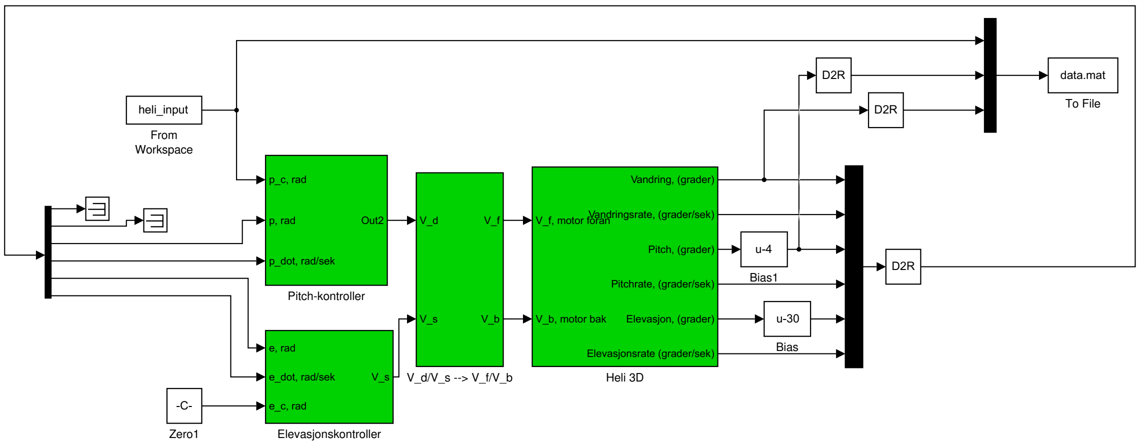

Figure 4: A Simulink diagram.

Bibliography