TTK4135 Optimization and Control
Helicopter Lab Report

716120, 723987
Group 1

April 29, 2015

Department of Engineering Cybernetics
Norwegian University of Science and Technology

Abstract

A movable arm capable of lateral and longitudinal motion by applied thrust from two attached rotors is stabilized by a basic regulatory layer with pitch and elevation angle PID controllers. The feasibility of controlling the arm to move from an initial equilibrium state to a given travel angle setpoint is investigated by employing in an open loop an optimal sequence of inputs for the PID controllers obtained by minimizing quadratic objective functions subject to linear as well as non-linear constraints. Enhanced tracking of the optimal travel trajectory is achieved by augmenting the basic control layer with a linear-quadratic controller.

Contents

1 Introduction

The work presented in this report reflects the assigned task of computing optimizing behavior of a movable arm actuated by two rotors. 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. Lastly, a nonlinear constraint is added to the optimization problem. Additional constraints are also investigated as a way of limiting nonlinear effects.

The report is organized as follows: Section 2 contains a closer description of the lab setup including prelimenary mathematical modeling. Section 3 describes PID-tuning of the plant as well as model parameter estimation to improve the mathematical models introduced in section 2. In section 4 open loop optimal control of the travel trajectory is introduced. Section 5 compares the results of augmenting the basic PID control layer with an LQR based on the optimal input and trajectories calculated in 4. In section 6 an optimal elevation trajectory is also considered, imposed with a non-linear inequality constraint yielding a non-linear optimization problem. Appendix A contains MATLAB scripts used in the work, where scripts used to generate data plots are omitted for brevity. Appendix B contains diagrams of the relevant Simulink models.

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:

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

$$\ddot{\lambda} = -K_2 p, \quad K_2 = \frac{K_p l_a}{J_t}, \tag{1b}$$

$$\ddot{e} = K_3 V_s - \frac{T_g}{J_e}, \quad K_3 = \frac{K_f l_a}{J_e}, \tag{1c}$$

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°.
- 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.
- The rotor thrust is assumed to be proportional to the voltage applied to the motor. This is a simplification. Generally, rotor angular velocity is proportional to the voltage applied, and thrust is proportional to the square of the angular velocity.

To stabilize the plant, adding to (1) 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 cancels out.

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

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 the measurement of the lab setup, and gives us the best model with the given number of states. This 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 Controller Tuning and Model parameter estimation

3.1 PID-tuning

The pre-tuned PID controllers showed unsatisfactory performance and was re-tuned to better serve as the stable plant for the rest of the assignment. In particular, the pre-tuned pitch controller was tuned in an aggressive fashion, impacting elevation control significantly. Figure 1 shows the coupling of pitch and elevation control before and after tuning. The two controllers were tuned independently of each other, in a manual fashion.

Table 3: Controller gains comparison

Gain	Original	Improved
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

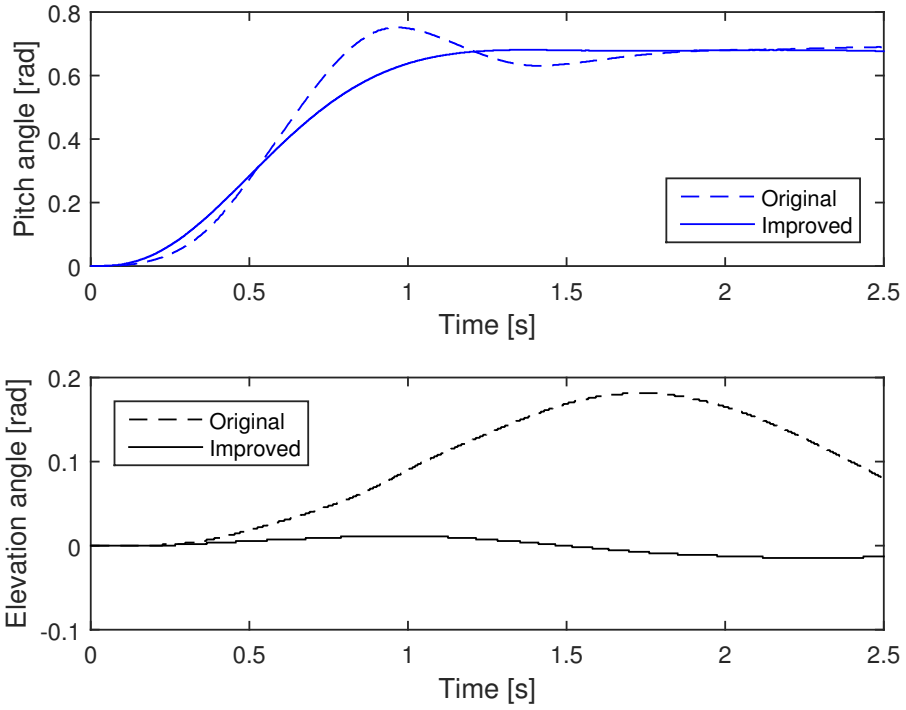

Figure 1: Pitch and elevation response to pitch step input, with original and improved pitch and elevation controller tuning.

3.2 Model parameter estimation

The derived model in (1) showed considerably different dynamics than the observed ones. The system identification toolbox in MATLAB was therefore used to compute the parameters which

fitted the recorded step responses. This was done by estimating the following three step responses based on known input and recorded system behavior:

- Pitch setpoint to pitch
- Elevation setpoint to elevation
- Pitch to travel rate

Figure 2 shows the pitch step response of the derived model compared with the newly estimated model, and the measured step response. The estimated model is clearly a better match than the derived model. Optimal input sequences calculated based on the derived model would need extensive help from feedback in order to yield usable performance. The elevation step responses of the derived and estimated model, shown in figure 3, and travel rate response in figure 4, displays similar results.

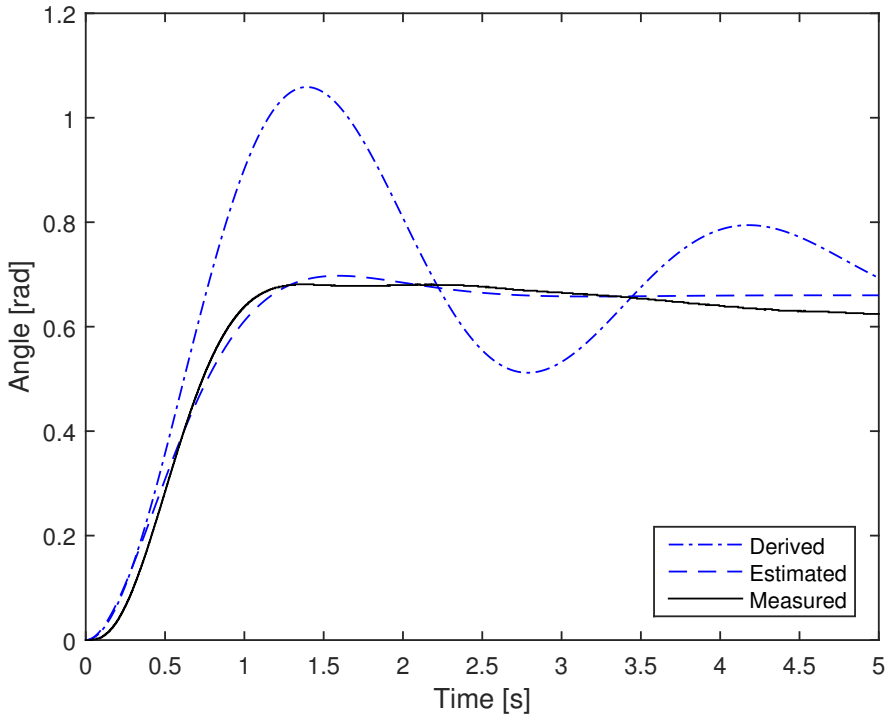

Figure 2: Pitch step responses of the derived and estimated model, compared with the measured step response.

3.3 Results and discussion

The following transfer functions were estimated from measured step responses:

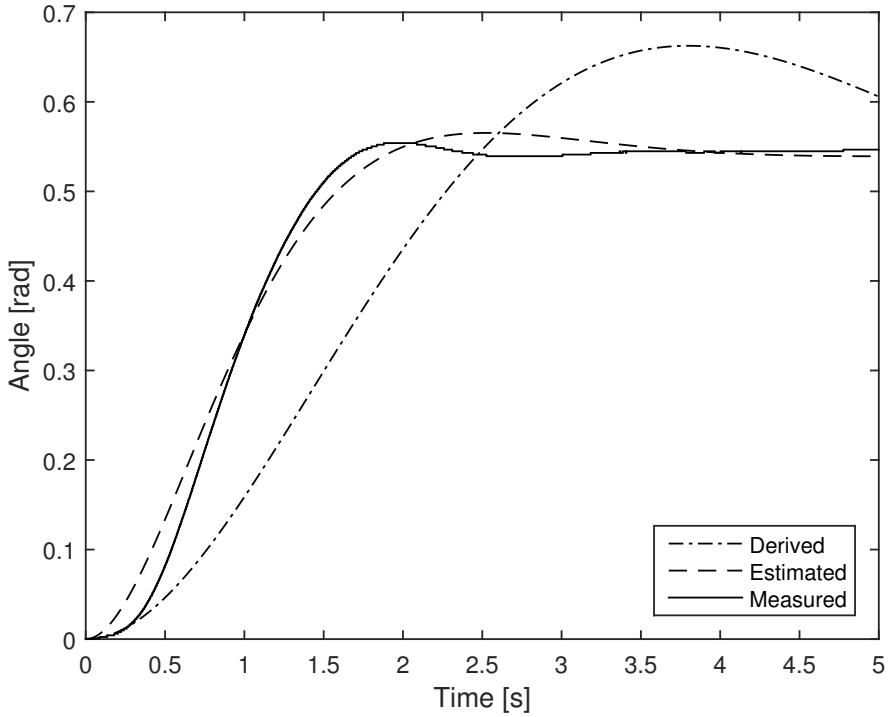

Figure 3: Elevation step responses of the derived and estimated model, compared with the measured step response.

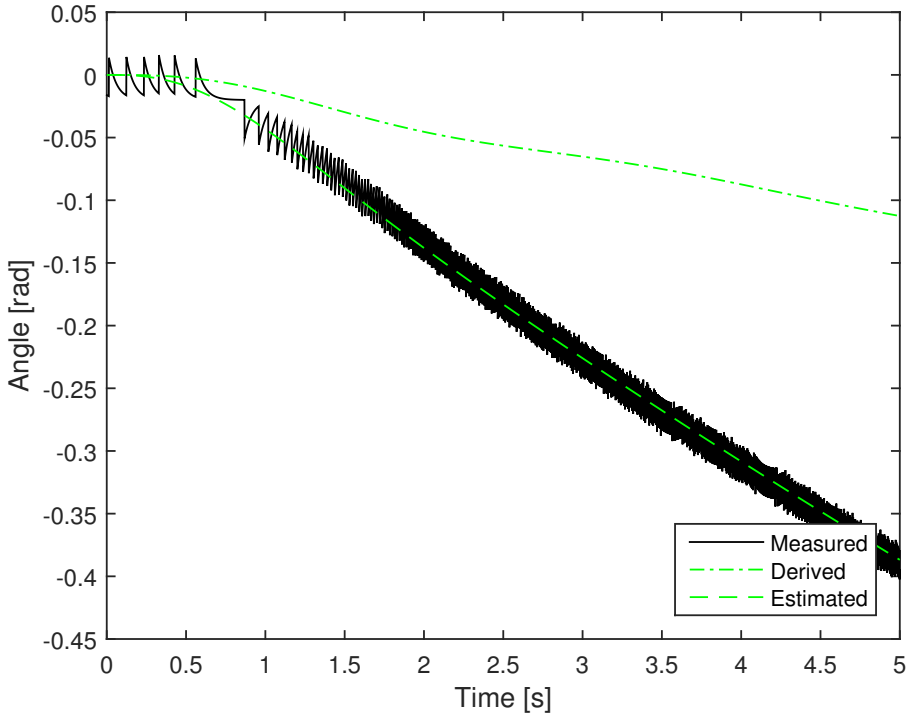

Figure 4: Travel rate response of commanded step in pitch. Derived and estimated model compared with the measured response.

$$\frac{p}{p_c}(s) = \frac{6.74}{s^2 + 3.60s + 7.13}, \tag{3a}$$

$$\frac{e}{e_c}(s) = \frac{3.13}{s^2 + 2.45s + 3.03}, \tag{3b}$$

$$\frac{\lambda}{p}(s) = \frac{-0.29}{s + 0.05}. \tag{3c}$$

The calculated response of the travel rate in figure 4 is almost identical to the recorded step response. The pitch and elevation models are satisfactory, but the dynamics are clearly not matching the actual dynamics to the same degree as with travel rate. This should be expected, as the small angle approximation used is bound to yield errors, especially in the pitch model. Both the off-axis hinged pitch head and the non-linearity of rotor thrust are also unaccounted for when settling on the number of states. A third state for both pitch and elevation might have helped capture some of the dynamics, but as we want to minimize the number of states, the results are found to be well within acceptable levels.

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 responses as discussed in 3.3. The model

$$\begin{bmatrix}\dot{\lambda}\\ \ddot{\lambda}\\ \dot{p}\\ \ddot{p}\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & -0.03 & -0.39 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & -7.13 & -3.6\end{bmatrix}}_{A_c}\begin{bmatrix}\lambda\\ \dot{\lambda}\\ p\\ \dot{p}\end{bmatrix} + \underbrace{\begin{bmatrix}0\\ 0\\ 0\\ 6.74\end{bmatrix}}_{B_c} p_c, \tag{4}$$

is constructed from (3) as well as some preliminary testing. The latter lead to a slight increase in the amount of change in travel rate as a function of pitch.

The reason for the increase in pitch's effect on travel rate is due to the fact that the pitch step response was run with a step size of 20°. The usual operating range, when aiming for somewhat aggressive maneuvers is around 30° to 45°. This amount of pitch angle will lead to a significant loss of downwards thrust and sequentially elevation. The elevation controller will in turn attempt to compensate by increasing the overall thrust by a large amount, but since the pitch head is far from equilibrium, a substantial amount of that thrust will be affecting the travel rate. This effect overshadows that of the small angle approximation.

4.2 Discretization

Let $x = \begin{bmatrix}\lambda & \dot{\lambda} & p & \dot{p}\end{bmatrix}^\top$, $u = p_c$. Using the Euler method[1] with a time step $\Delta t = 0.25$s we are able to obtain an approximate discretization of (4):

$$\begin{aligned} x_{k+1} &= x_k + \Delta t \dot{x}_k \\ &= x_k + \Delta t(A_c x_k + B_c u_k) \\ &= (I + \Delta t A_c)x_k + (\Delta t B_c)u_k \\ &= Ax_k + Bu_k, \end{aligned} \tag{5}$$

[1] As described in Egeland and Gravdahl (2003).

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

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

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

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

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

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 (5) subjects (7) to the linear equality constraints

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

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

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

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

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. With rapid convergence rate in mind the pitch bounds were set to $\pm\frac{45\pi}{180}$. The optimal input sequence u^* is applied to the plant in an open loop with results shown in figure 5, with the measured trajectory compared to the calculated optimal trajectory x^*.

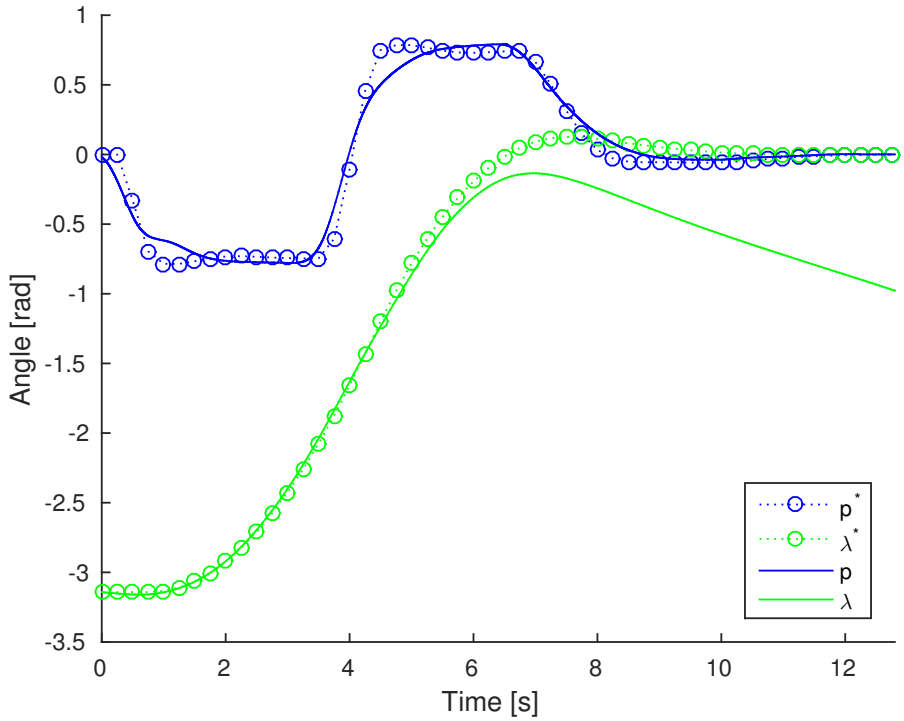

Figure 5: Optimal pitch and travel trajectories compared with measured trajectories with optimal input sequence applied in an open loop.

The measured trajectory of pitch coincides with the calculated optimal trajectory. The lack of substantial deviation is also a testimonial to the pitch model. Unlike pitch, travel is not controlled by an inner controller, and its deviation from the optimal trajectory should therefore be expected when using open loop.

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

5.1 Discrete-time LQR

To eliminate the discrepency between the optimal and measured travel trajectory observed in figure 5, we augment the optimal input for every time step with a state feedback term weighted by a suitable gain matrix K:

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

or, alternatively

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

where

$$\Delta x_k = x_k - x_k^*,$$
$$\Delta u_k = u_k - u_k^*.$$

It can be shown[2] that the controller (10) is the optimal solution minimizing the quadratic objective function

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

subject to the system dynamics (5), 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 in (10), and the resulting Linear-quadratic regulator is implemented, with weigthing matrices $\tilde{Q}$ and $\tilde{R}$ chosen to penalize deviations in states and input for satisfactory results.

5.2 Results and discussion

The state penalty matrix $\tilde{Q}$ and input penalty $\tilde{R}$

$$\tilde{Q} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{R} = 0.1.$$

are chosen with travel accuracy in mind. Deviation in travel as well as travel rate is penalized fairly hard. Input deviation is necessary for corrections in travel and travel rate, and a low penalty is therefore chosen.

Using MATLAB's `dlqr` we obtain the LQ state feedback gain K. Using the optimal input sequence u^* and state trajectory x^* obtained from solving (9) the controller is applied with results shown in figure 6.

[2] Its optimality is derived in numerous books, e.g. Kwakernaak and Sivan (1972).

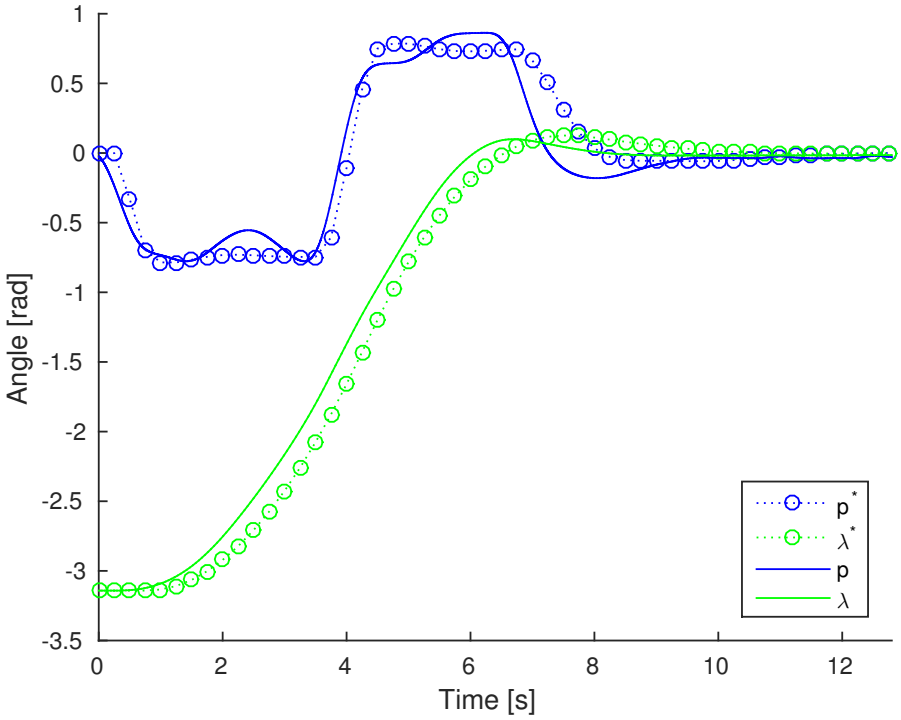

Figure 6: Optimal pitch and travel trajectories compared with measured trajectories with optimal input sequence applied in an LQR closed loop.

The measured trajectories are similar to the ones without feedback, shown in figure 5, except for the removal of travel drift over time. The travel rate model is clearly not perfect as the measured trajectory seemingly wants to deviate from the optimal path, but the discrepancies are small and the state feedback controller deals with them nicely.

5.2.1 MPC discussion

An alternative approach to optimal trajectories with state feedback would be to use an MPC controller. An MPC controller simply computes the optimal trajectory from the current state to the target state given a finite window of time. Only the first input of the resulting optimal input sequence is used. This is then repeated at the next iteration[3]. Solving an optimization problem at each time iteration is clearly computationally expensive, and even QP problems of this relatively small size would force the iteration period to increase substantially. An MPC controller with a time step on the order of several seconds is unlikely to yield better performance than a predetermined trajectory with short time steps combined with state feedback control, due to the fast dynamics of our system. Deviation from optimal trajectory is also likely to be caused by either an incomplete model or unmodeled external disturbances, and an MPC controller will not withstand either very well.

[3]Description based on Foss and N. Heirung (2014).

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

6.1 State space model

Adding e and $\dot{e}$ to the previous state space model (4) using (3b), we get

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

The added states are clearly decoupled from the rest of the system. This is due to the fact that the model is linearized around the equilibrium. At small pitch angles this simplification does not cause any large discrepancies. However, when the pitch angle is large, an increase in elevation rate is clearly accompanied by an increase in travel rate.

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 the Euler method we obtain a discrete state space model

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

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

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

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 an inequality constraint is imposed 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 (13) 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

$$A_{eq} z = B_{eq}, \tag{17b}$$
$$c(x_k) \leq 0, \quad k = \{1, \ldots N\}, \tag{17c}$$
$$p^{\text{low}} \leq p_k, p_{c_{k-1}} \leq p^{\text{high}}, \quad k = \{1, \ldots N\}. \tag{17d}$$
$$\tag{17e}$$

6.4 Discrete-time LQR

In addition to running the optimal input sequence u^* in an open loop, a discrete-time LQR as defined in 5.1 is applied, with the weigthing matrices

$$\tilde{Q} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \tilde{R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{18}$$

6.5 Optional: Additional constraints

Although the calculated input yielded satisfactory performance, the model has shortcomings. Specifically, when a slwe impose ow descent in elevation is commanded the rotor blades almost come to a complete stop, during which control of pitch is severely reduced. As this is impossible to model with a linear model, additional bounds are imposed on elevation rate. This will reduce

the effect of the unmodeled coupling of elevation to the rest of the system. In a further attempt to keep the system within the linear area the travel rate is also bounded:

$$-0.05 \leq \dot{e_k} \leq 0.05, \quad k = \{1, \dots N\},$$
$$-0.5 \leq \dot{\lambda_k} \leq 0.5, \quad k = \{1, \dots N\}.$$

6.6 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` using an active set method. The pitch bounds are also tightened from 45° to 25° in order to lower the effect of elevation rate and travel rate coupling at large pitch angles.

The optimal input sequence u^* is applied to the plant in an open loop with results shown in figure 7. The system follows the optimal trajectory except for the usual discrepancy in travel present when using open loop control. We further employ the LQR to guide the trajectory to the set-point and eliminate steady state deviations. The result, shown in figure 8, is without any large discrepancies between optimal and measured trajectory. However, the constraint bound imposed on elevation is violated, as shown in figure 10. The reason for this is probably a combination of issues. The optimal path is calculated with a fairly large time step, in order to lower the computational time required. This leads to an optimal path that when interpolated will violate the constraint. Also, as the increase in elevation will largely happen when the pitch angle is at its maximum, and the resulting increase in elevation will be lower than if the increase were to be commanded at zero pitch. State feedback is also unable to deal with this problem completely, as any aggressive elevation correction is bound to happen at the cost of pitch, and therefore travel rate control. Lastly, 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, as visible in figure 8, effectively will shift the optimal elevation path with respect to travel. This lag is visible in figure 10.

The mentioned effects contributing to constraint violation are attempted minimized by the addition of extra constraints on elevation rate and travel rate, as mentioned in 6.5. The resulting optimal path reflects this, and the slope of the optimal elevation path is significantly reduced. This yields an overall less aggressive system, with decreased violation of the elevation constraint. The added constraint on travel rate also means that the system uses the entire time slot to reach the desired end state.

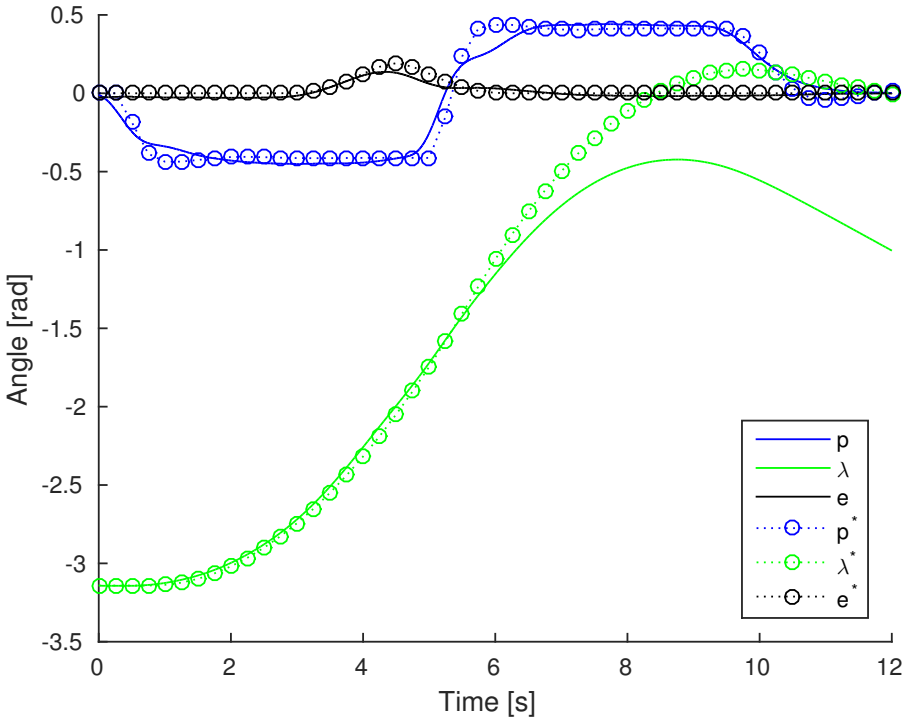

Figure 7: Optimal pitch, travel and elevation trajectories compared with measured trajectories, with optimal input sequence applied in an open loop.

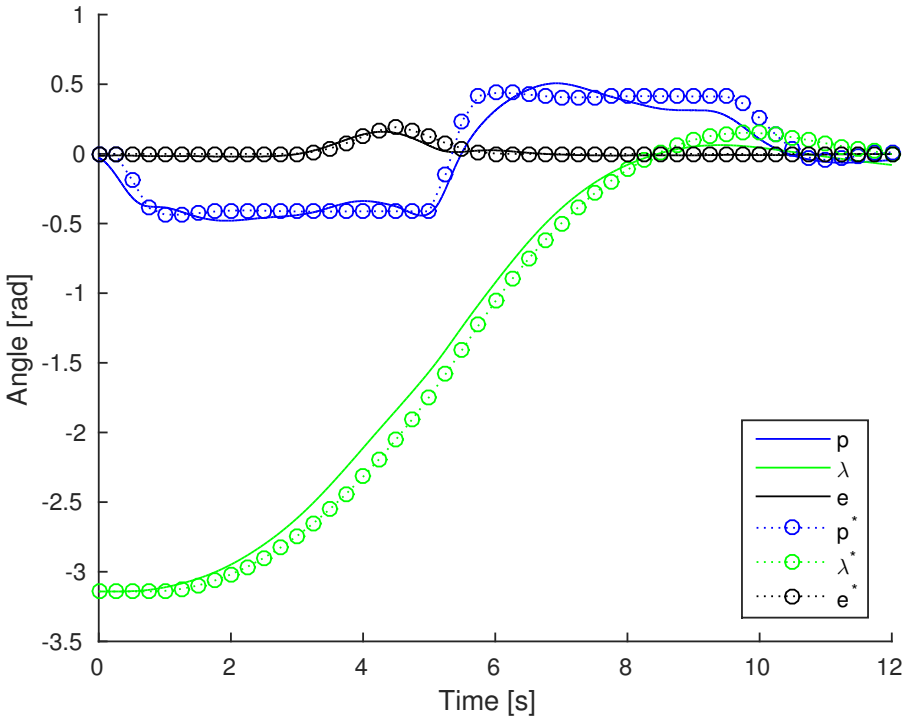

Figure 8: Optimal pitch, travel and elevation trajectories compared with measured trajectories, with optimal input sequence applied in an LQR closed loop.

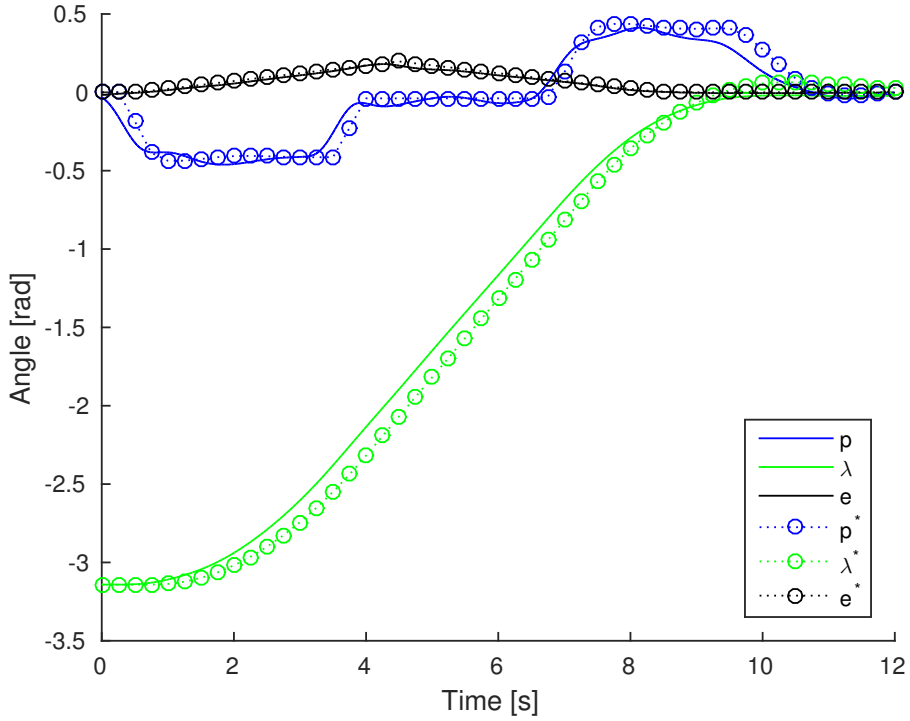

Figure 9: Optimal trajectories compared with closed loop measured trajectories, with lower and upper bounds added to elevation rate and travel rate.

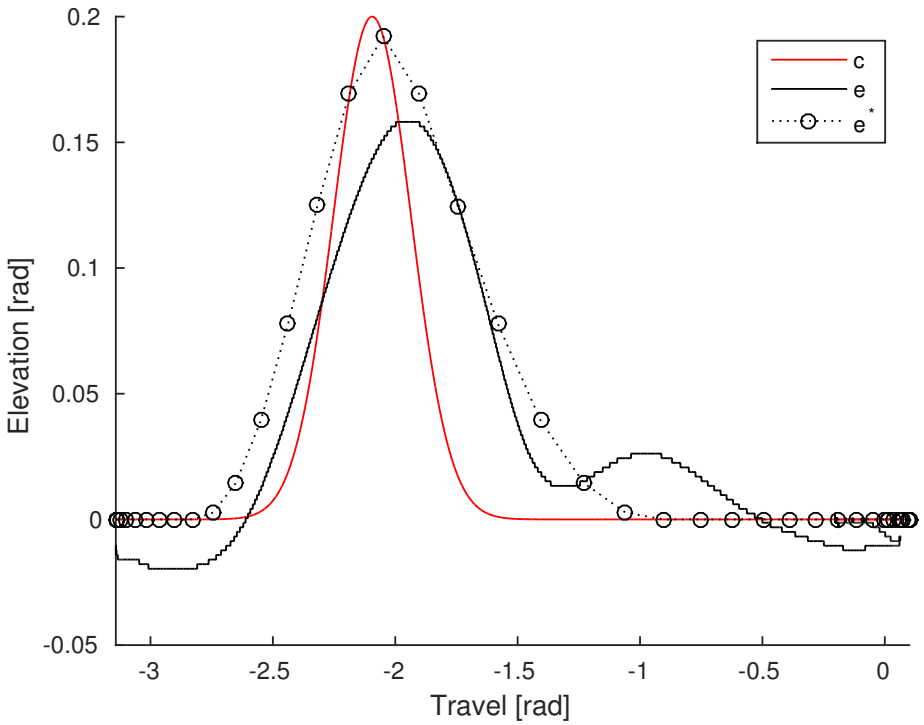

Figure 10: Inequality contraint imposed on elevation compared with the optimal and measured trajectories, with LQR.

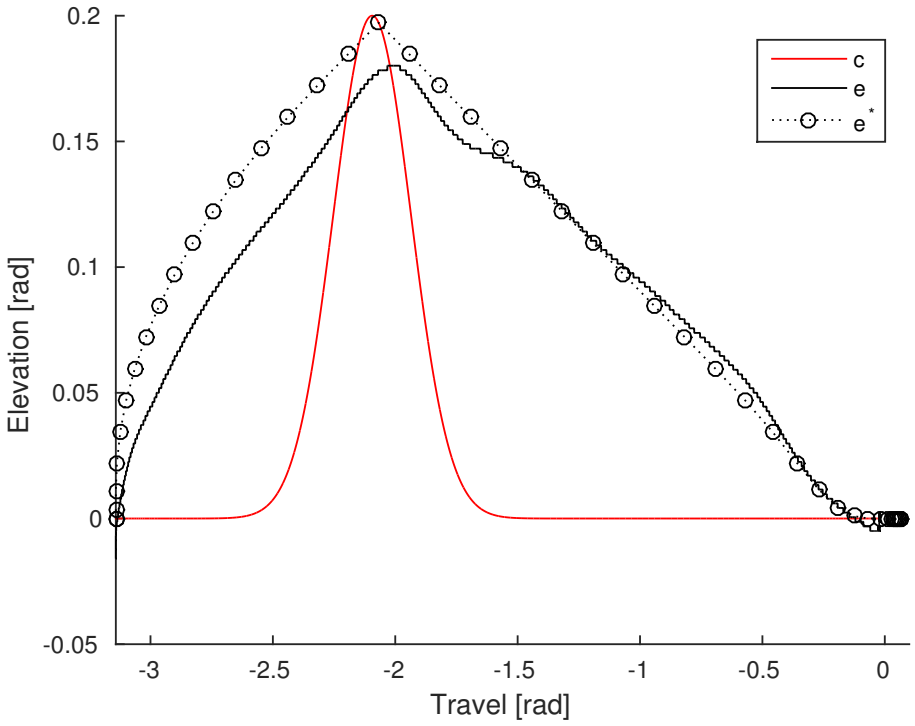

Figure 11: Inequality contraint imposed on elevation compared with the optimal and measured trajectories, with additional lower and upper bounds on elevation rate and travel rate.

7 Discussion

The optimal paths computed have shown to be quite compatible with the actual system. This is reflected in the displayed data. The reason for discrepancies have also been relatively easy to identify. This is probably due to mainly two related facts. The controllers of pitch and elevation were tuned in a way that gave the least nonlinear coupling between them. This greatly improves the changes of a linear model being sufficient. Secondly, the statistically system identification yields a model with an accuracy far greater than one based on indirect measurements. Optimal paths based on an incomplete system model will probably yield suboptimal performance at best.

As shown in both figure 5 and 7 the need for feedback control is present. A linear quadratic controller is a computationally inexpensive way of achieving this. The feedback importance is also reduced by having an accurate system model, and only light corrections were necessary.

The effect of model shortcomings as discussed in 3.3, 4.1, 6.1, 6.5 and 6.6 are attempted minimized by adding extra constraints to the optimization problem. This forces the system to operate inside the linear area, yielding slower but more accurate performance.

8 Conclusion

Optimal operation of the movable arm is achieved by minimizing a quadratic cost function, and precomputed input sequences are supplemented by state feedback. The linear model is shown to yield satisfactory performance, and enhanced performance is achieved by adding constraints designed to minimize nonlinear effects.

A MATLAB Code

A.1 System identification

```
clear all; close all; clc

%% 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

%% Elevation
% h = elevation/elevation_ref

load elevStep30deg
u_elev = 30*pi/180 * ones(7000,1);
u_elev(1) = 0;
y_elev = elevStep30deg.signals.values(1:7000) + 16.8*pi/180; % unbias the step
elev_data = iddata(y_elev, u_elev, 0.001);
elev_time = 0.001:0.001:7;

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

%% 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
```

A.2 Optimal control of pitch/travel without feedback

```
clear all; close all; clc
init02

%% Continuous model
Ac = [0    1    0    0;
      0    -0.03 -0.39   0;
      0    0    0    1;
      0    0  -7.13  -3.6];
Bc = [ 0 0 0 6.74]';

%% Discrete model
dt = 0.25;
A = eye(4) + Ac*dt;
```

```
B = Bc*dt;

% given: x0 = [0 0 0 0]';
xf = [pi 0 0 0]';

n_x = size(A,2);
n_u = size(B,2);

%% Simulatin parameters
duration = 25;
N = floor(duration/dt);
r = .1;
pitch_lim = 45; % deg

%% Equality constraints

Aeq = [ eye(N*n_x) + kron(diag(ones(N-1,1),-1), -A) , kron(eye(N), -B)];

beq = [-A*xf;
       zeros(n_x*(N-1),1)];

%% Bounds
LB_x = repmat([-Inf -Inf -pitch_lim*pi/180 -Inf]', N, 1);
UB_x = repmat([Inf Inf pitch_lim*pi/180 Inf]', N, 1);

LB_u = repmat(-pitch_lim*pi/180, N, 1);
UB_u = repmat(pitch_lim*pi/180, N, 1);

LB = [LB_x;
      LB_u];

UB = [UB_x;
      UB_u];

%% Quadratic objective function
Q = zeros(n_x);
Q(1,1) = 1;

R = r;

G = blkdiag(kron(eye(N), Q), kron(eye(N), R));

%% Solve QP
[z,fval,exitflag,output,lambda] = quadprog(G, [], [], [], Aeq, beq, LB, UB);

x = reshape(z(1:N*n_x), [n_x, N]);
travel_opt = [-xf(1), x(1,:)];
pitch_opt = [-xf(3), x(3,:)];
u = [reshape(z(N*n_x+1:end), [n_u, N]) , zeros(n_u, 1)];

%% Prep input sequence
padding_time = 10;
padded_input = [zeros(1,floor(padding_time/dt)) , u]';
time = [(0:length(padded_input) - 1)*dt]';
heli_input = [time padded_input];
```

A.3 Optimal control of pitch/travel with feedback (LQ)

```
%% LQR
Q = diag([4,2,0,0]);
```

```
R = 0.1;
K = dlqr(A,B,Q,R);

%% Prep reference trajectory
x_opt = x + repmat(xf, 1, length(x)); % Shift travel ref by pi
padded_x_opt = [zeros(4,floor(padding_time/dt)) , x_opt];
time = [(0:length(padded_x_opt) - 1)*dt];
heli_ref = [time; padded_x_opt]';
```

A.4 Optimal control of pitch/travel and elevation with and without feedback

```
clear all; close all; clc
init02

%% Continuous model
Ac = [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    0 -3.03  -2.44];

Bc = [ 0    0;
       0    0;
       0    0;
       6.74 0;
       0    0;
       0    3.13];

%% Discrete model
dt = .25;
A = eye(6) + Ac*dt;
B = Bc*dt;

% given: x0 = [0 0 0 0 0 0]';
xf = [pi 0 0 0 0 0]';

n_x = size(A,2);
n_u = size(B,2);

%% Simulation parameters
duration = 12;
N = floor(duration/dt);
r1 = .1;
r2 = .1;
pitch_lim = 25; % deg
elev_lim = 50;
elev_rate_lim = 0.05; % Inf
travel_rate_lim = 0.5; % Inf

%% Equality constraints

Aeq = [ eye(N*n_x) + kron(diag(ones(N-1,1),-1), -A) , kron(eye(N), -B)];

Beq = [-A*xf;
       zeros(n_x*(N-1),1)];

%% Bounds
LB_x=repmat([-Inf -travel_rate_lim -pitch_lim*pi/180 -Inf -elev_lim*pi/180 -elev_rate_lim ]',N,1)
UB_x=repmat([Inf travel_rate_lim pitch_lim*pi/180 Inf elev_lim*pi/180 elev_rate_lim]',N,1);
```

```
50
51  LB_u = repmat([-pitch_lim*pi/180 -elev_lim*pi/180]', N, 1);
52  UB_u = repmat([pitch_lim*pi/180 elev_lim*pi/180]', N, 1);
53
54  LB = [LB_x;
55        LB_u];
56
57  UB = [UB_x;
58        UB_u];
59
60
61  %% Quadratic objective function
62  Q = zeros(n_x);
63  Q(1,1) = 1;
64
65  R = [r1 0;
66       0  r2];
67
68  G = blkdiag(kron(eye(N), Q), kron(eye(N), R));
69
70  f = @(X) X'*G*X;
71
72  %% Solve optimization problem
73  tic
74  [X, FVAL, EXITFLAG] = fmincon(f, zeros(N*8,1), [], [], Aeq, Beq, LB, UB, @constraint);
75  toc
76
77  x = reshape(X(1:N*n_x), [n_x, N]);
78  travel_opt = [-xf(1), x(1,:)];
79  pitch_opt = [-xf(3), x(3,:)];
80  elevation_opt = [-xf(5), x(5,:)];
81  u = [reshape(X(N*n_x+1:end), [n_u, N]) , zeros(n_u, 2)];
82
83  %% LQR
84  Q = diag([4,2,0,0,3,0]);
85  R = diag([1 1]);
86
87  K = dlqr(A,B,Q,R); % Closed loop
88  %K = zeros(2,6);     % Open loop
89
90  %% Prep input sequence
91  padding_time = 10;
92  padded_input = [zeros(2,floor(padding_time/dt)) , u]';
93  time = [(0:length(padded_input) - 1)*dt]';
94  heli_input = [time padded_input];
95
96  %% Prep reference trajectory
97  x_opt = x + repmat(xf, 1, length(x)); % Shift travel ref by pi
98  padded_x_opt = [zeros(6,floor(padding_time/dt)) , x_opt];
99  time = (0:length(padded_x_opt) - 1)*dt;
100 heli_ref = [time; padded_x_opt]';
```

B Simulink Diagrams

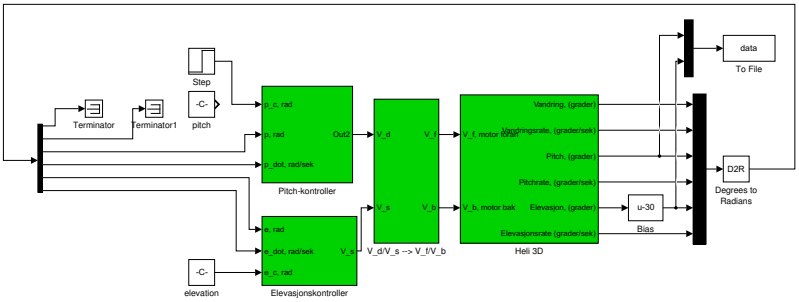

Figure 12: Simulink model used in section 3.

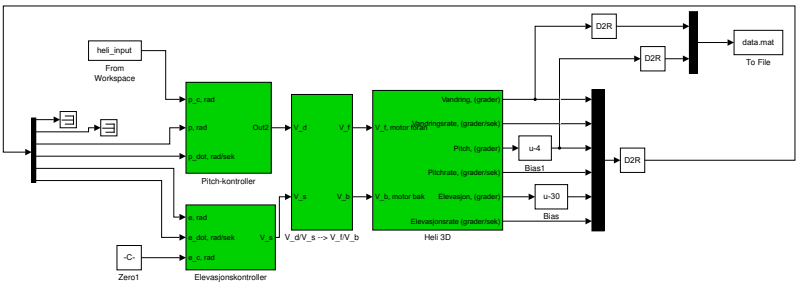

Figure 13: Simulink model used in section 4.

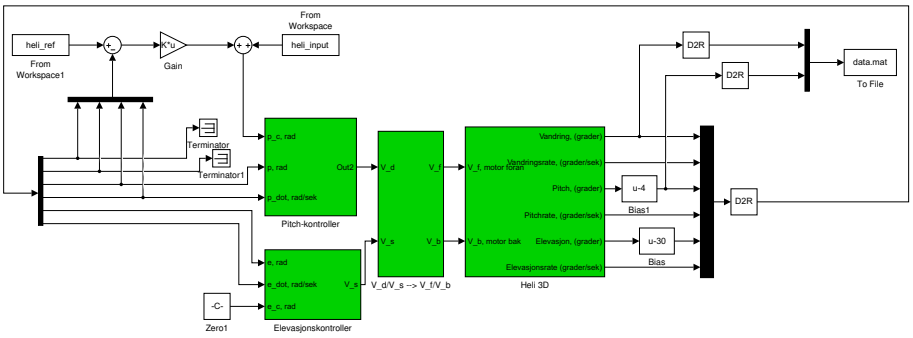

Figure 14: Simulink model used in section 5.

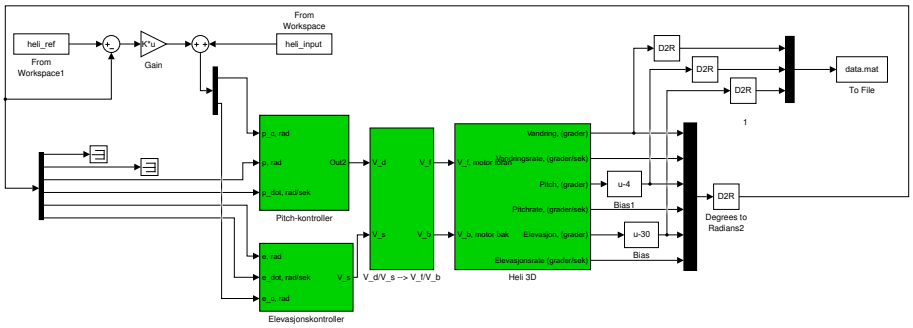

Figure 15: Simulink model used in section 6.

Bibliography

Egeland, O. and Gravdahl, T. (2003). *Modeling and Simulation for Automatic Control*. Marine Cybernetics.

Foss, B. and N. Heirung, T. A. (2014). Merging optimization and control.

Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems. First Edition*. Wiley-Interscience.