NTNU

Norwegian University of
Science and Technology

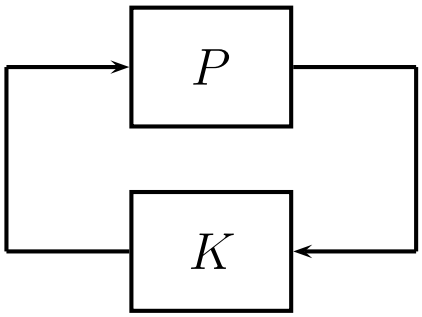
STUDIES IN STATIC OUTPUT FEEDBACK CONTROL

Master's thesis

Submission date: July 2009

Author:
Bjarne Grimstad

Supervisors:
Professor Ole Morten Aamo
Professor Sigurd Skogestad
PhD fellow Henrik Manum



Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

This master's thesis studies the recently proposed initialization scheme for the $\mathscr{H}_2$-optimal static output feedback problem [Manum et al., 2009]. The initialization scheme consists of solving a convex quadratic programming (QP) problem to obtain the controller (K_0) that in the open-loop sense is closest to the linear quadratic regulator (LQR). For the special case when the available measurements $y_k = Cx_k$, do not contain all information about the system states, we have that the resulting control law is $u_k = -KC^{\dagger}y_k = -K_0 y_k$, where K is the LQR. In this thesis we show that for a class of systems this controller is suitable for initializing the $\mathscr{H}_2$-optimal static output feedback problem. In particular, we have tested the syntesized $\mathscr{H}_2$-optimal controllers on the thermal/optical plant uDAQ28/LT [Jelenčiak et al., 2007]. A short study of the convexity properties of the $\mathscr{H}_2$-optimal static output feedback problem has shown that the problem is non-convex for some systems. Thus, we cannot expect to find the globally $\mathscr{H}_2$-optimal static output feedback for cases where the initialization procedure fails to provide a good controller guess.

Preface

This thesis was written at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU), spring 2009. The project assignment was specified and written by Sigurd Skogestad and Henrik Manum at the Department of Chemical Engineering (IKP), NTNU.

The assignment is part of an ongoing research in the process control group at IKP, NTNU. The group is trying to develope the theory of self-optimizing control by extending it, building branches that connect it with acknowledged fields such as model predictive control. It has been a great motivation for me to know that I have a chance to participate in the research and maybe influence it by sharing my opinions. Hopefully, my master's thesis contributes to the building of one of these branches.

I would like to thank professor Sigurd Skogestad and Henrik Manum for their commitment and for always taking their time to answer my questions. Their insight in control engineering has been a motivation alone. I would like to give a special thanks to Henrik Manum who has helped me extensively throughout the process of writing this thesis. His understanding and assistance have been much appreciated. I also gratefully acknowledge the financial support I have received from IKP to realize my two visiting trips to the Swiss Federal Institute of Technology (ETH), Zurich.

Not to be forgotten; I have to grant my boys at the office some credit for always being at the office... preventing me from growing lonely in those dark hours at the end of this semester. I also owe Jakub Osusky from the Slovak University of Technology, Bratislava, a favor for providing me with information about the uDAQ28/LT, and for doing it with a smile:)

Contents

Abstract **i**

Preface **iii**

List of Figures **vii**

List of Tables **x**

List of Abbreviations **xii**

1 Introduction **1**
1.1 Motivation & goals 3
1.2 Document structure 4
1.3 Code library 4

2 Theoretical background **5**
2.1 Static Output Feedback 6
2.2 PID control 10
2.3 Linear Quadratic Optimal Control 12
2.4 Self-optimizing Control 24

3 Review of literature **31**
3.1 Introducing the problem 31
3.2 Finding the optimal measurement combination H 32
3.3 Closed-loop optimization 36
3.4 Initialization of the $\mathscr{H}_2$-optimal static output feedback problem . . 38

4 Convexity of the $\mathscr{H}_2$-optimal static output feedback problem **39**
4.1 Convexity of optimization problems 40
4.2 Examples 42
4.3 Discussion 47

5 Examples **49**
5.1 $\mathscr{H}_2$-optimal control of a second-order system 50

5.2 Study of the thermal/optical plant uDAQ28/LT 57
5.3 Robustness properties of LQG 77

6 Final discussion **81**

7 Conclusion **85**
7.1 Summary 86
7.2 Future research 87

References **89**

A Linear system theory **93**
A.1 The generalized plant 93
A.2 Linear fractional transformations 94

B Derivations **97**
B.1 Truncation of prediction horizon 97
B.2 Deriving J_{uu} and J_{ud} 101
B.3 Approximation of the $\mathscr{H}_2$-optimal static output feedback problem . 105
B.4 Model augmentation 107

C Proof of convexity **111**
C.1 SISO systems with a single state 112
C.2 The general case – Finding an approach 114

D Simulink schemes **115**
D.1 Robustness properties of LQG: Simulink schemes 115
D.2 Thermal/Optical Plant: Simulink schemes 117

List of Figures

1.1 Feedback control of the plant G. 1

2.1 The Kalman filter loop. 17
2.2 The Kalman filter and noisy plant. 18
2.3 Illustration of the separation theorem. 20
2.4 LQG with integral action. 21
2.5 The LQG problem formulated in the general control configuration. . . . 23
2.6 Feedback implementation of optimal operation with separate layers for optimization (RTO) and control. 25

4.1 Example of a convex and non-convex function. 41
(a) A convex function 41
(b) A non-convex function 41
4.2 Frequency response of $G(s)$. 43
4.3 The $\mathscr{H}_2$ objective function $J_{cl} = ||F_l(P,K)||_2$ as a function of K. 44
4.4 The $\mathscr{H}_2$ objective function $J_{cl} = ||F_l(P,K)||_2$ as a function of K. 45
4.5 Approximation of the $\mathscr{H}_2$ norm for different horizon lengths. 46
(a) Approximations of the $\mathscr{H}_2$ norm. 46
(b) A close-up of the figure at the left. 46

5.1 Output resulting from a disturbance in the initial state ($\zeta = 0.8$). 53
(a) Closed-loop responses for $\zeta = 0.8$. 53
(b) Close-up of the responses for $\zeta = 0.8$. 53
5.2 Plot of the $\mathscr{H}_2$-optimal PID controller gains as a function of ζ. 54
(a) $\mathscr{H}_2$-optimal PID controller gains. 54
(b) Bandwidth and margins. 54
5.3 Output resulting from a disturbance in the initial state ($\zeta = 0.5$ and $\zeta = 0.2$). 55
(a) Closed-loop responses for $\zeta = 0.5$. 55
(b) Closed-loop responses for $\zeta = 0.2$. 55
5.4 Plots of the $\mathscr{H}_2$ norm (closed-loop objective) for $\zeta = 0.5$ and $\zeta = 0.2$ and PI controller. 56
(a) $||F_l(P,K)||_2$ for $\zeta = 0.5$. 56

(b) $||F_l(P,K)||_2$ for $\zeta = 0.2$. 56
5.5 The uDAQ28/LT 57
5.6 Schematic drawing of the uDAQ28/LT. 58
5.7 Frequency-dependent RGA for *model 1* and *model 2*. 61
(a) RGA elements for *model 1*. 61
(b) RGA elements for *model 2*. 61
5.8 Measurement sequence of temperature (y_1) and light intensity (y_2) around steady-state. 62
5.9 PSD of measurement sequences. 63
(a) PSD of temperature measurement. 63
(b) PSD of light measurement. 63
(a) Frequency response of the Equiripple high-pass filter. 64
(b) High-pass filtered temperature measurement. 64
5.10 Temperature and light intensity measurements from the *udaq* using LQG, PID, and PI control (*model 1*). 71
(a) Temperature measurement (LQG) 71
(b) Light intensity measurement (LQG) 71
(c) Temperature measurement (PID) 71
(d) Light intensity measurement (PID) 71
(e) Temperature measurement (PI) 71
(f) Light intensity measurement (PI) 71
5.11 Temperature and light intensity measurements from the *udaq* using LQG, PID, and PI control (*model 2*). 72
(a) Temperature measurement (LQG) 72
(b) Light intensity measurement (LQG) 72
(c) Temperature measurement (PID) 72
(d) Light intensity measurement (PID) 72
(e) Temperature measurement (PI) 72
(f) Light intensity measurement (PI) 72
5.12 Controller outputs for the two cases (*model 1* and *model 2*) of LQG, PID, and PI control of the *udaq*. 73
(a) LQG controller outputs (*model 1*) 73
(b) LQG controller ouputs (*model 2*) 73
(c) PID controller outputs (*model 1*) 73
(d) PID controller outputs (*model 2*) 73
(e) PI controller outputs (*model 1*) 73
(f) PI controller outputs (*model 2*) 73
5.13 Kalman filter estimation error with LQG control of the *udaq*. 74
(a) Estimation error with *model 1* 74
(b) Estimation error with *model 2* 74
5.14 The maximum singular value of the sensitivity function with LQG, PID, and PI control of *model 2*. 74
5.15 Nyquist diagram of $G_{ol}(s)$ 79
5.16 Simulation of LQG controlled plant without time delay. 80

(a) Measurement and output estimate 80
(b) Input usage 80
5.17 Simulation of LQG controlled plant with a time delay of 0.08 s. 80
(a) Measurement and output estimate 80
(b) Input usage 80

A.1 General control configuration 93
A.2 Illustration of lower and upper LFT 95
(a) R_l as lower LFT in terms of K_l 95
(b) R_u as upper LFT in terms of K_u 95

D.1 Simulink scheme of noisy plant and LQG controller. 115
D.2 Observer scheme (continuous steady-state Kalman filter). 116
D.3 Simulink scheme of the three controllers and thermal/optical plant. . . 117
D.4 Simulink scheme showing the internals of the thermal plant subsystem. 118
D.5 Simulink scheme of the LQG controller. The controller consists of a discrete steady-state Kalman filter and a LQR. 118
D.6 Simulink scheme of the PID controller. 119
D.7 Simulink scheme of the low-pass filtered PI controller. 119

List of Tables

1.1 Folder structure 4

2.1 Notation 16
2.2 The discrete Kalman filter equations. 16
2.3 Notation 25

3.1 Main algorithm – Low-order controller synthesis 38

5.1 Weight matrices 51
5.2 Controllers for $\zeta = 0.8$ 52
5.3 Controllers for $\zeta = 0.5$ 53
5.4 Controllers for $\zeta = 0.2$ 54
5.5 Global optimality? 55
5.6 Plant inputs 57
5.7 Plant outputs 57
5.8 *udaq* models. 60
5.9 Description of RGA elements. 61
5.10 Noise properties for the *udaq* measurements. 64
5.11 Weight matrices for *model 1* 66
5.12 Weight matrices for *model 2* 67
5.13 Objective function values for *model 1* and *model 2*. 68

List of Abbreviations

LQG	Linear Quadratic Gaussian
LQR	Linear Quadratic Regulator
MIMO	Multi-Input-Multi-Output
MPC	Model Predictive Control
PID	Proportional-Integral-Derivative
RHP	Right Half-Plane
SISO	Single-Input-Single-Output
SOF	Static Output Feedback

Chapter 1

Introduction

When N. Wiener developed the field of *cybernetics* in 1948 and wrote the book with the same title, he did it on the basis of feedback control [Wiener, 1948]. Simple feedback loops, often using proportional-integral-derivative (PID) control, has been a vital part of any control engineer's toolbox ever since. The theory behind this powerful tool is known as "classical control" and is documented in the works of Bode, Nichols, and others from the 1940's.

S. Skogestad points out *simplicity*, *robustness*, and *stabilization* as three fundamental advantages of feedback control [Skogestad, 2009]. *Simplicity* in that tight control can be achieved even with a very crude model. *Robustness* since feedback is required for making a system adapt to new conditions. And *stabilization*; because feedback is the only way to fundamentally change the dynamics of a system. E.g. feedback is the only way to stabilize an unstable plant.

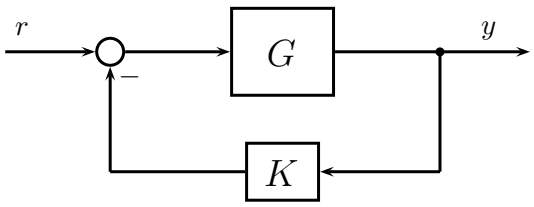


Figure 1.1: Feedback control of the plant *G*.

The virtues of feedback are lessened when the control loop is subject to a large phase lag (time delay). Phase lags introduced by unstable (RHP) zeros and "effec-

tive time delays" limit the controller gain and the advantages of a high feedback gain diminishes. These fundamental limitations apply for any controller and cannot be avoided even with sophisticated model-based control, as we will see later in this thesis.

Even though feedback from output measurements has been used for a long time there still exist unsolved problems, the *static output feedback problem* probably being the most important [Syrmos et al., 1997]. The unanswered question is; when can we use a static output feedback to obtain some desired closed-loop characteristic? Many years of research on the problem have given several important results, but not yet a satisfactory answer. The reason that this problem has been granted so much attention is that it involves a big class of problems. Also, it is desirable because static output feedback is the simplest of all feedback implementations, not requiring state estimation and generally not a very exact model.

The static output feedback problem may be studied in the setting of linear quadratic (Gaussian) control. Unlike the conventional formulation, where the quadratic objective is minimized by choosing future plant inputs, the optimization problem is then solved with the controller parameters as the degrees of freedom. The resulting optimization problem for finding the optimal static output feedback may also be cast as an $\mathscr{H}_2$-optimal control problem, which is the preferred formulation in this thesis.

The work presented in this thesis is mainly based on the recent research on a convex initialization scheme for the $\mathscr{H}_2$-optimal static output feedback problem [Manum et al., 2009]. The proposed initialization scheme is reviewed and tested on some interesting systems, including the thermal/optical plant uDAQ28/LT. A qualitative analysis of the convexity of the $\mathscr{H}_2$-optimal static output feedback problem is also given.

1.1 Motivation & goals

Model-based control has enjoyed a lot of attention from the control community the last decade or so, model-predictive control (MPC) probably being the most popular approach. Today MPC is considered a mature alternative to classical control, e.g. PID feedback. The inherently multivariable formulation and constraint handling are two reasons for the success of MPC. However, for large-scale systems there still exist challenges in form of increased computational demands, and today a lot of research is being done on improving optimization efficiency (for example by model and problem reduction). In addition, MPC generally requires a state estimator, usually implemented as a Kalman filter. On the other hand, a static output feedback solution does not require a state estimator, nor much (online) computational effort. Thus, for large systems static output feedback could prove itself to be a real alternative to model predictive control with state estimation.

In this thesis we aim to compare model-based control, in form of LQG control, with the static output feedback approach. We will use the initialization procedure in [Manum et al., 2009] to synthesize low-order $\mathscr{H}_2$-optimal controllers for several interesting systems. Especially we want to test the procedure on the thermal/optical plant uDAQ28/LT where noise is present and should be taken into account in the controller synthesis. Further, we wish to investigate the convexity properties of the optimization problem used to find the $\mathscr{H}_2$-optimal static output feedback controllers. The goal with this investigation is to learn more about static output feedback and what problems that may occur with this approach.

The scope of this thesis is limited to linear time-invariant systems without input, output, or state constraints. The hope is to raise some new questions that can fuel an ongoing research.

1.2 Document structure

This thesis is divided into three parts, outlined here. The first part is a literature study. The literature study is separated into two chapters, the first of those holds some basic control theory while the following goes more into depth and presents a review of [Manum et al., 2009]. In the second part a problem statement is given before the contributions resulting from the work on this thesis is presented. The thesis is ended by a final discussion and a conclusion drawn on the knowledge obtained through the study. Finally, a list of points regarding future research and some hints on what it may bring is provided.

This document holds several appendices with theory, derivations, and figures that the author has split from the main text for increased reading pleasure.

1.3 Code library

The theory, algorithms, and examples in this thesis were implemented in Matlab™ code. Table 1.1 shows the folder structure of the developed code library. The code is attached to this document.

Table 1.1: Folder structure

Folder name	Description
./	Root folder
./clopt/	Algorithm for closed-loop optimization.
./optmeas/	Implementation of the theory from self-optimizing control.
./examples/	Holds the code for all the examples in this thesis.
./lib/	External libraries.

Chapter 2

Theoretical background

This chapter holds some basic control theory which is needed in the next chapters of this master's thesis. We will first present the static output feedback problem and list some of the most important results that has been obtained through the many years of research on the difficult topic. We then move on to the more pleasant topics of PID and LQG control. These topics are well-developed and widely accepted. Thus, we only consider the key elements of them and aspects relevant in later chapters. Next, we acquaint the theory and idea of self-optimizing control. This theory is central in the review chapter to come, where the main algorithm in this thesis is presented. The reader is advised to read the theoretical background in this chapter before moving onwards.

We restrict our discussion to linear, time-invariant (LTI) systems which in continuous time can be formulated as

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{2.1a}$$

$$y(t) = Cx(t) + Du(t), \tag{2.1b}$$

where $x \in \mathbb{R}^{n_x}$ is the *state* vector, $u \in \mathbb{R}^{n_u}$ is the *input* vector, and $y \in \mathbb{R}^{n_y}$ the *output* vector. $A \in \mathbb{R}^{n_x \times n_x}$ is the state matrix, $B \in \mathbb{R}^{n_x \times n_u}$ is the input matrix, $C \in \mathbb{R}^{n_y \times n_x}$ is the output matrix, and $D \in \mathbb{R}^{n_y \times n_u}$ is the direct feedthrough matrix. The discrete counterpart of (2.1) is

$$x_{k+1} = Ax_k + Bu_k, \tag{2.2a}$$

$$y_k = Cx_k + Du_k, \tag{2.2b}$$

where the dimensions are the same as in the continuous case and the subscript k refers to time instants.[1]

[1] To ease notation the time dependance is often omitted, i.e. $x(t) = x$ or $x_k = x$. It should be clear from the context what is meant in each specific case.

2.1 Static Output Feedback

The static output feedback (SOF) problem is arguably one of the most important open questions in control engineering; see for example [Blondel et al., 1995]. This section provides a short summary of the SOF problem and is mainly based on the clear survey by Syrmos et al. [Syrmos et al., 1997]. A general formulation of the problem is:

> Given a linear time-invariant system, find a static output feedback so that the closed-loop system has some desirable characteristics, or determine that such a feedback does not exist.

Here "desirable characteristic" could be pole placement for example. Stated differently, the problem of pole placement with static output feedback is to find a

$$u = -K^y y \tag{2.3}$$

which places the poles (or the eigenvalues) of the closed-loop system $\dot{x} = (A - BK^yC)x$ as desired. A system is said to be (output) *pole-assignable* if all poles may be assigned to a desirable set of poles. We say that the output feedback is static, meaning that the feedback gain K^y is a constant matrix (vector or scalar).

The fact that many control problems can be reduced to some variation of the SOF problem strengthens its importance. E.g. the design of a dynamical output compensator of order $q \leq n$, where n is the system order, may be brought back to the static output feedback case as shown in [Syrmos et al., 1997]. Modern control design techniques, such as LQG and MPC with an observer, provides controllers of order equal to or greater than the order of the plant ($q \geq n$). Hence avoiding the SOF problem. However, for large-order systems these controllers are difficult or impossible to implement owing to cost, reliability, and hardware limitations. This motivates the search for simple, low-order controllers and reveals the SOF problem.

The literature gives several necessary and sufficient conditions for the two problems of stabilizability and pole placement with static output feedback. Some of them are listed below, starting with a few important (but untestable) conditions for pole placement via static output feedback.

Necessary condition 1. A necessary condition for generic pole assignability with a real gain matrix K^y is that $n_u n_y \geq n_x$.

Sufficient condition 1. If (A,B,C) is minimal (i.e. controllable and observable) with B and C of full rank then $\max(n_u, n_y)$ poles are assignable.

Sufficient condition 2. $n_u n_y > n$, where n is the McMillan degree of the system, is sufficient for generic pole assignability [Wang, 1996].

We proceed by presenting the *even PIP* theorem and a necessary condition for stabilization of an open-loop unstable plant by static output feedback.

Theorem 2.1 (Even PIP)**.** *A linear system $H(s)$ is stabilizable with a stable controller $C(s)$ that has no real unstable zeros* iff *(i) the number of real poles of $H(s)$, counted according to their McMillan degree, between any pair of real blocking zeros in the right half-plane is even and (ii) the number of real blocking zeros of $H(s)$ between any two real poles of $H(s)$ is even. In this case we say that $H(s)$ satisfies the even* parity-interlacing property *(even PIP).*

Necessary condition for static output stabilizability. A necessary condition for static output stabilizability of an open-loop unstable plant $H(s)$ is that it satisfy the even PIP.

Example 1 The following example from [Syrmos et al., 1997] illustrates the necessary condition cited above. Consider the plant

$$H(s) = \frac{1-s}{(\epsilon s+1)(s-2)}. \tag{2.4}$$

For $\epsilon > 0$, the even PIP is not satisfied, and the plant is not SOF-stabilizable. For $-1 < \epsilon \leq 0$, the even PIP is satisfied. However, a simple root-locus analysis shows that this plant is SOF-stabilizable only for $-0.5 \leq \epsilon \leq 0$. This short example underlines that the even PIP is a necessary, but not sufficient condition for output stabilizability.

The problem of finding stabilizing static output controllers has been approached from many angles; the inverse linear-quadratic approach, covariance assignability by output feedback, output structural constraint approach, and decision methods, to mention a few [Syrmos et al., 1997]. Unfortunately, the conditions resulting from these approaches are either too restrictive, not efficiently testable, or both. Moreover, the computational algorithms for testing these conditions are highly complex and not even guaranteed to converge. To better understand why this problem is so hard to solve we consider another two approaches, namely the "Lyapunov approach" and "LQ approach". We allow ourselves to copy these from [Syrmos et al., 1997].

The Lyapunov approach. Necessary and sufficient conditions for static output feedback can be obtained in terms of coupled linear matric inequalities following a quadratic Lyapunov-function approach. From Lyapunov stability theory we know that the (continuous) closed-loop system matrix $A - BK^yC$ is stable iff K^y satisfies the following matrix inequality:

$$(A - BK^yC)P + P(A - BK^yC)^\top < 0 \tag{2.5}$$

for some $P > 0$. For a fixed P, the inequality (2.5) is a linear matrix inequality (LMI) in the matrix K. The LMI (2.5) is convex in K, so that convex programming can be used to numerically find a K whenever $P > 0$ is given. Conditions for static output feedback stabilization are obtained by finding the solvability conditions of (2.5) in terms of K [Iwasaki et al., 1994].

Theorem 2.2. *There exists a stabilizing static output feedback gain iff there exists a* $P > 0$ *such that*

$$B^\perp(AP + AP^\top)(B^\perp)^\top < 0 \tag{2.6}$$

$$(C^\top)^\perp(A^\top P^{-1} + P^{-1}A)[(C^\top)^\perp]^\top < 0 \tag{2.7}$$

where $B^\perp$ *and* $(C^\top)^\perp$ *are full-rank matrices, orthogonal to* B *and* $C^\top$ *respectively.*

Notice that (2.6) is an LMI on P and (2.7) is an LMI on P^{-1}.[2] Computational methods based on iterative sequential solutions of the two convex LMI problems with respect to P and P^{-1} have been proposed to find stabilizing static output feedback gains, but convergence of the algorithms is not guaranteed [Iwasaki et al., 1994].

The LQ approach. The SOF problem may also be studied in a linear quadratic regulator (LQR) setting. Consider the following optimization problem:

$$\min_{K^y} J = \int_0^\infty x^\top Qx + u^\top Ru\,dt \tag{2.8}$$

$$\text{s.t. } \dot{x} = (A - BK^yC)x, \tag{2.9}$$

with $Q \geq 0$ and $R > 0$. In addition, it is required that K^y stabilizes the closed-loop system. The necessary conditions for optimality were given in [Levine & Athans, 1970] and are repeated beneath:

$$0 = A_c^\top S + SA_c + Q + C^\top K^{y\top} RK^yC \tag{2.10}$$

$$0 = A_cP + PA_c^\top + X \tag{2.11}$$

$$0 = RK^yCPC^\top - B^\top SPC^\top \tag{2.12}$$

[2] It is interesting to note that many other static output feedback control problems, such as suboptimal $\mathscr{H}_\infty$ control, suboptimal linear quadratic control and μ synthesis with constant scaling can be formulated in terms of coupled LMIs as in (2.6) and (2.7), see for example [Iwasaki & Skelton, 1994].

with $X = x(0)x(0)^\top$ and $A_c = A - BK^yC$. Generally, optimal control with reduced information results in such coupled nonlinear matrix equations. The dependence of (2.10)-(2.12) on the initial state $x(0)$ may be removed by setting $X = E\{x(0)x(0)^\top\}$. E.g. it is common to assume that $x(0)$ is uniformly distributed on the units sphere, so that $X = I$.

> Conditions for the existence and global uniqueness of solutions to (2.10)-(2.12) such that P and S are positive definite and (2.9) is stable are not known. It has been shown that in the discrete case there exists a gain that minimizes (2.8) locally and also stabilizes the system if $Q \geq 0$, $R > 0$, rank$(C) = n_y$, $X > 0$, and (A,B,C) is output stabilizable. However, there may be more than one local minimum, so that the solution of (2.10)-(2.12) may not yield the global minimum. [Syrmos et al., 1997]

Iterative methods for finding K^y has been proposed, see for example [Moerder & Calise, 1985]. However, these algorithms guarantee only a local minimum and require the selection of an initial stabilizing gain. A direct procedure for finding such a K^y is unknown as discussed in [Syrmos et al., 1997].

To summarize; few SOF design techniques or solution algorithms are available today. The excisting algorithms are too restrictive, suffer from untestable conditions (transforms the problem into another unsolved problem), and/or is computational inefficient. The result is that the existing methods are characterized as *ad hoc* solutions.

Complexity analysis of the problem of stabilizability by static output feedback has indicated that it is $\mathcal{NP}$-hard, that is, equally hard to solve as any decision problem in the complexity class $\mathcal{NP}$ [Syrmos et al., 1997]. The negative result that the *(0,1)-Knapsack problem* [Cormen et al., 2001] can be reduced to the problem of pole placement via static output feedback in polynomial time came in a recent publication [Fu, 2004]. Thus, pole placement with static output feedback is proved to be $\mathcal{NP}$-hard. This implies that moderately large problems are computationally intractable. Syrmos et al. [Syrmos et al., 1997] concludes the survey by suggesting that every effort should be made on exploiting the special structure of each particular problem.

2.2 PID control

The proportional-integral-derivative (PID) controller is the most commonly used control algorithm in industrial control systems [Skogestad & Postlethwaite, 2005]. A common implementation of the PID controller is the parallel (or ideal) form

$$K_{pid}(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right), \tag{2.13}$$

where the parameters are the gain K_p, integral time T_i, and derivative time T_d. The PID control law is $u(s) = K_{pid}(s) \cdot e(s)$, where $e = r - y$ is the error and r is the desired output (often called reference signal or setpoint). The controller is not realizable on this form since it is not proper. To obtain a proper controller it is common to include limited derivative action by filtering the derivative part of the controller[3], that is

$$u(s) = K_c\left[1 + \frac{1}{T_i s} + \frac{T_d s}{\epsilon T_d s + 1}\right] e(s). \tag{2.14}$$

The filter characteristic is chosen by $\epsilon = (0, 1]$. Setting $\epsilon = 0$ gives back the pure derivative, however with this value the system is not realizable. With ϵ close to zero the controller becomes sensitive to noise, because the noise is differentiated. To increase robustness against process variations and lower sensitivity to noise it is common to roll-off the controller gain at high frequencies. This can be achieved by additional low-pass filtering of the control signal, i.e.

$$K_{pid}(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \cdot \frac{1}{(T_f s + 1)^n}, \tag{2.15}$$

where T_f is the filter time constant and n is the order of the filter.

The PID controller is tuned by selecting parameters K_p, K_i, and K_d, that give an acceptable closed-loop response. A desirable response is often characterized by the measures of settling time, oscillation period, and overshoot, to mention a few. In addition, one has to ensure that the PID controller has good robustness properties. The robustness properties may be quantified by the gain margin and phase margin, which essentially tells us how close the closed-loop system is to instability. We know from Bode's stability criterion that the loop gain must be less than 1 at the critical frequency ω_{180} where the phase lag around the loop is -180 degrees (-360 degrees including the negative gain in the feedback loop). Otherwise, signals at this frequency will increase in magnitude for each pass through the loop and we have instability.

[3] In practical implementations the reference signal is usually not differentiated. This is to avoid derivative kick when a step occur in the reference signal. With this implementation the controller has two degrees of freedom because the signal path from y to u is different from that from r to u.

Many PID tuning methods have been proposed over the years; ranging from the simple, but most famous Ziegler-Nichols tuning method [Ziegler & Nichols, 1942], to the more modern simple internal model control (SIMC) tuning rules by Skogestad [Skogestad, 2003]. Various authors have also suggested PID tuning methods based on the LQR, e.g. [He et al., 2000]. The PID controller can also be tuned via loop-shaping techniques, where the loop transfer functions is shaped by changing the controller parameters.[4]

The PID controller can be generalized to the multi-input multi-output (MIMO) case where $K_{PID}(s)$ becomes a full transfer function matrix, connecting each input-output pair with a single-input single-output (SISO) PID controller. In this case we call $K_{PID}(s)$ a *MIMO PID controller*. The parallel-form PID controller with limited derivative action (2.14) can be generalized to the MIMO case by

$$K_{pid}(s) = \left[K_p + K_i \frac{1}{s} + K_p \frac{s}{\epsilon s + 1} \right], \tag{2.16}$$

where K_p, K_i, and K_p are $n_u \times n_y$-matrices. On this form, all input-output pairs share the same low-pass filter on the derivative action, and the controller in (2.16) is a special case of the general MIMO PID controller. This limitation may affect control when roll-off (of the derivative) is desired at different frequencies for the input-output pairs. However, in most cases ϵ is given a small value (typically $\epsilon = 0.1$) for all input-output pairs. Another special case is when K_p, K_i, and K_d are square and diagonal matrices, we then have decentralized control, i.e. each input is paired with one output.

Control systems are today realized in computer software which demand a discrete implementation. One way to obtain the discrete version of the PID controller is to use the bilinear transform:

$$s \leftarrow \frac{2}{T_s} \frac{z-1}{z+1}, \tag{2.17}$$

where T_s is the sampling time. The bilinear transform is a first-order approximation of the natural logarithm function that is an exact mapping of the z-plane to the s-plane (Z-transform) [Balchen et al., 2003].

[4] It is quite ironic that even with all the research that has been layed down in finding controller parameters that in some sense is optimal, most PID controllers running in the industry run on factory settings.

2.3 Linear Quadratic Optimal Control

In this chapter we will go through some properties of optimal control of unconstrained LTI systems with a quadratic cost function. The scope of this chapter is confined to the deterministic linear regulator problem, also known as the LQR problem, the Kalman filter, and the LQG problem. Lastly, we will talk about $\mathscr{H}_2$-optimal control and the link between it and the LQG problem. No attempt is made to give an in-depth analysis of these well-known control problems, and we merely point out some of their most important characteristics. The reason for looking at these problems is that they constitute the foundation of the work we present later in this thesis.

The discussion is restricted to linear, time-invariant (LTI) systems with noise, i.e.

$$\dot{x}(t) = Ax(t) + Bu(t) + w_d(t), \tag{2.18a}$$
$$y(t) = Cx(t) + w_n(t), \tag{2.18b}$$

where we have set $D = 0$ for simplicity. $w_d(t)$ and $w_n(t)$ are the process and measurement noise, in the following we assume them to be uncorrelated zero-mean Gaussian stochastic processes with constant power spectral density matrices W and V, respectively. In other words we have that

$$E\left\{w_d(t)\,w_d(t)^\top\right\} = W\delta(t-\tau), \tag{2.19a}$$
$$E\left\{w_n(t)\,w_n(t)^\top\right\} = V\delta(t-\tau), \tag{2.19b}$$

and

$$E\left\{w_d(t)\,w_n(t)^\top\right\} = 0, \tag{2.20a}$$
$$E\left\{w_n(t)\,w_d(t)^\top\right\} = 0, \tag{2.20b}$$

where $E\{\cdot\}$ is the expectation operator and $\delta(t-\tau)$ is a delta function. In the discrete case we will use the same notation for the process and measurement noise (w_d and w_n) and the power spectral density matrices (W and V). However, we underline here that W and V in continuous time in general are different from W and V in discrete time. See [Brown & Hwang, 1997] for the relation.

2.3.1 Continuous-time Linear Quadratic Regulator

The LQR problem, where all states are known, can be stated as follows: given the system $\dot{x} = Ax + Bu$ with a non-zero inital state $x(0)$, find the input $u(t)$ which takes the system to the zero state ($x = 0$) in an optimal manner, i.e. by minimizing the cost

$$J_r = \int_0^\infty x(t)^\top Q x(t) + u(t)^\top R u(t)\, dt. \tag{2.21}$$

The optimal solution (for any inital state) is

$$u(t) = -K_r x(t), \tag{2.22}$$

where the feedback gain matrix is derived by

$$K_r = R^{-1} B^\top P, \tag{2.23}$$

$$0 = PA + A^\top P - PBR^{-1}B^\top P + Q. \tag{2.24}$$

Eq. (2.24) is a continuous-time algebraic Ricatti equation (CARE) and has an unique positive semidefinite solution $P = P^\top \geq 0$.

2.3.2 Discrete-time Linear Quadratic Regulator

In discrete time the infinite-horizon LQR problem is to minimize the cost function

$$J_r = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k, \tag{2.25}$$

subject to the system constraint $x_{k+1} = Ax_k + Bu_k$. The optimal solution is a state feedback $u_k = -K_r x_k$ where K_r is found by solving

$$K_r = R^{-1} B^\top A^\top (P - Q), \tag{2.26}$$

$$P = A^\top P (I + BR^{-1}B^\top P)^{-1} A + Q. \tag{2.27}$$

Here, Eq. (2.27) is a discrete-time algebraic Ricatti equation (DARE) with an unique positive semi-definite solution $P = P^\top \geq 0$.

2.3.3 Comments on the Linear Quadratic Regulator

Lets address briefly some features of the infinite-horizon LQR problem in Eqs. (2.21) and (2.25), starting with the issue of stability.

The optimal solution to the infinite-horizon LQR problem, $u = -K_r x$, gives an asymptotically stable system if the pair (A, B) is *stabilizable* and $(A, Q^{\frac{1}{2}})$ is *detectable* [Naidu, 2003]. We can interpret these requirements as follows: If we have unstable states we must be able to detect them via the cost function in order to control them. Hence we require that $(A, Q^{\frac{1}{2}})$ is detectable, i.e. all unstable states are observable. Further, we must require that (A, B) is stabilizable, i.e. all unstable states are controllable, to guarantee that we can stabilize the unstable states.

The weight matrices are often assumed to be constant and diagonal, and we require that $Q \geq 0$ and $R > 0$. In order to keep the integral/sum of the expressions $x^\top Qx$ in the cost function small and non-negative, we understand that Q must be *positive semidefinite*. Due to the quadratic nature of the weightage, large errors (here $e = x - 0 = x$) are penalized more than small errors. Also, from $u^\top Ru$ we see that one has to pay a higher cost for larger control effort. Since the cost of the control has to be a positive quantity, that is, we do not want to award zero control action, the R matrix should be *positive definite*. Time-varying weights are covered in [Naidu, 2003].

The weight matrices can be considered as the tuning parameters of the LQR by regarding Q as a state (error) penalty, and R as a penalty on input usage or input power. Alternatively, we can penalize the outputs (y) by chosing $Q = C^\top Q_y C$, where Q_y is the output penalty. Loosely speaking, the factor $||Q||/||R||$ can be chosen to favourize either fast control or cheap control (low input power).

The LQR has several important advantanges:

- It is inherently multivariable (takes care of coupling in the process).
- It is optimal on the infinite horizon.
- It has good robustness properties (it can be non-robust when implemented in combination with state estimation).

However, the LQR does not handle constraints. When constraints on states and/or inputs are added the quadratic programming (QP) problem has infinite optimization variables and becomes intractable. Luckily, the infinite-horizon problem can

be approximated with a finite-horizon (length N) cost function

$$J_r = \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \approx x_N^\top S x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k, \tag{2.28}$$

where S is the *terminal cost* on the final state x_N. When S satisfies the Ricatti equation (2.27) (or (2.24) in the continuous case), the relation is exact and the finite-horizon problem also provides the solution to the *unconstrained* infinite-horizon LQR problem. However, when putting structure on the allowed input moves (constraints), optimizing on the finite horizon is not necessarily the same as optimizing on the infinite horizon, unless the horizon is "long enough". Several techniques exist to compute a "long enough" horizon, that is, a horizon N that guarantee stability and feasibility of the QP problem, see for example [Keerthi & Gilbert, 1988], [Gilbert & Tan, 1991].

The choice $S = P$ (Ricatti matrix) implies that the LQR law ($u = -K_r x$) is utilized after the horizon. Another option is to choose $K_r = 0$ after the horizon and S as the solution of the discrete Lyapunov equation $S = A^\top SA + Q$.[5] The latter being meaningful only for open-loop stable systems. The matter of truncating the horizon is discussed in more detail in Appendix B.1.

[5] The Lyapunov equation becomes $A^\top P + PA + Q = 0$ in the continuous case. It is then called a continuous Lyapunov equation.

2.3.4 The Discrete Kalman Filter

The Kalman filter is probably the most well-known and widely used tool for stochastic estimation from noisy measurements. It is named after Rudolph E. Kalman, who in 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem [Kalman, 1960].

The Kalman filter is an optimal observer that estimates the states from noisy measurements by a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of noisy measurements. It is optimal in that it minimizes the error covariance $P_k = E\{[x_k - \hat{x}_k][x_k - \hat{x}_k]^\top\}$ by selecting an optimal Kalman gain K_k at each time instant. It operates recursively in a loop consisting of two distinct phases: *time update* and *measurement update*.

Table 2.1: Notation

$\hat{x}_{k\|k-1}$	The *a priori* estimate of x_k given measurements up to and including y_{k-1}
$\hat{x}_{k\|k}$	The *a posteriori* estimate of x_k given measurements up to and including y_k
$P_{k\|k-1}$	The *a priori* error covariance, $P_{k\|k-1} = E\{[x_k - \hat{x}_{k\|k-1}][x_k - \hat{x}_{k\|k-1}]^\top\}$
$P_{k\|k}$	The *a posteriori* error covariance, $P_{k\|k} = E\{[x_k - \hat{x}_{k\|k}][x_k - \hat{x}_{k\|k}]^\top\}$

Using the notation in Table 2.1, the Kalman filter for the discrete LTI system

$$x_{k+1} = Ax_k + Bu_k + w_{d,k}, \tag{2.29}$$

$$y_k = Cx_k + w_{n,k}, \tag{2.30}$$

with covariance matrices $E\{w_d w_d^\top\} = W$ and $E\{w_n w_n^\top\} = V$, consists of the equations summarized by Table 2.2.

Table 2.2: The discrete Kalman filter equations.

Time update	Measurement update
$\hat{x}_{k\|k-1} = A\hat{x}_{k-1\|k-1} + Bu_{k-1}$ $P_{k\|k-1} = AP_{k-1\|k-1}A^\top + W$	$K_k = P_{k\|k-1}C^\top(CP_{k\|k-1}C^\top + V)^{-1}$ $P_{k\|k} = (I - K_kC)P_{k\|k-1}$ $\hat{x}_{k\|k} = \hat{x}_{k\|k-1} + K_k(y_k - C\hat{x}_{k\|k-1})$

After initialized with an initial state x_0 and covariance matrix P_0 the recursive Kalman filter proceeds in a loop consisting of the time and measurement update. The loop is illustrated by Figure 2.1. The *time update* is responsible for projecting forward in time the current state and error covariance estimates to obtain the *a priori* estimates for the next time step. The measurement update is responsible for the feedback – i.e. for incorporating new measurements into the *a priori* estimate to obtain an improved *a posteriori* estimate.

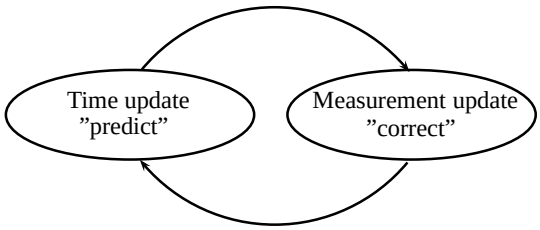


Figure 2.1: The Kalman filter loop.

The error covariance P_k and Kalman gain K_k generally converge as time goes on, and a suboptimal Kalman filter can be implemented by replacing them with their respective steady-state values. Lets denote the steady-state values $P_k = P_{k+1} = P$ and $K_k = K_{k+1} = K_f$. The suboptimal Kalman filter is then obtained by substituting P and K_f into the equations in Table 2.2. This is the *steady-state Kalman filter*.

The steady-state Kalman gain K_f, also known as the innovation gain, can be found off-line by solving

$$K_f = PC^\top (CPC^\top + V)^{-1}, \tag{2.31}$$

where $P = P^\top \geq 0$ is the unique positive semidefinite solution of the discrete algebraic Ricatti equation (DARE)

$$P = AP(I + PC^\top (CPC^\top + V)^{-1} CP)A^\top + W. \tag{2.32}$$

The equations for $\hat{x}_{k|k-1}$ and $\hat{x}_{k|k}$ in Table 2.2 can then be combined into one time-invariant state-space model (the steady-state Kalman filter)

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + AK_f(y_k - C\hat{x}_k), \tag{2.33a}$$
$$\hat{y}_k = C\hat{x}_k + CK_f(y_k - C\hat{x}_k). \tag{2.33b}$$

From (2.33) we see that the Kalman filter has the same order as the system with eigenvalues at $eig(A - AK_fC)$. The reader is referred to [Brown & Hwang, 1997] and [Kalman, 1960] for more information about the properties and underlying assumptions of the Kalman filter.

2.3.5 The Continuous Kalman Filter

The continuous steady-state Kalman filter has the structure of an ordinary state estimator or observer, as shown in Figure 2.2. For the system in (2.18) the estimator is

$$\dot{\hat{x}} = A\hat{x} + Bu + K_f(y - C\hat{x}), \tag{2.34}$$

$$\hat{y} = C\hat{x}. \tag{2.35}$$

The optimal choice of K_f minimizes $E\{[x-\hat{x}]^\top [x-\hat{x}]\}$ and is given by

$$K_f = PC^\top V^{-1}, \tag{2.36}$$

where $P = P^\top \geq 0$ is the unique positive semidefinite solution of the continuous algebraic Ricatti equation (CARE)

$$PA^\top + AP - PC^\top V^{-1} CP + W = 0, \tag{2.37}$$

with W and V defined by Eqs. (2.19) and (2.20).

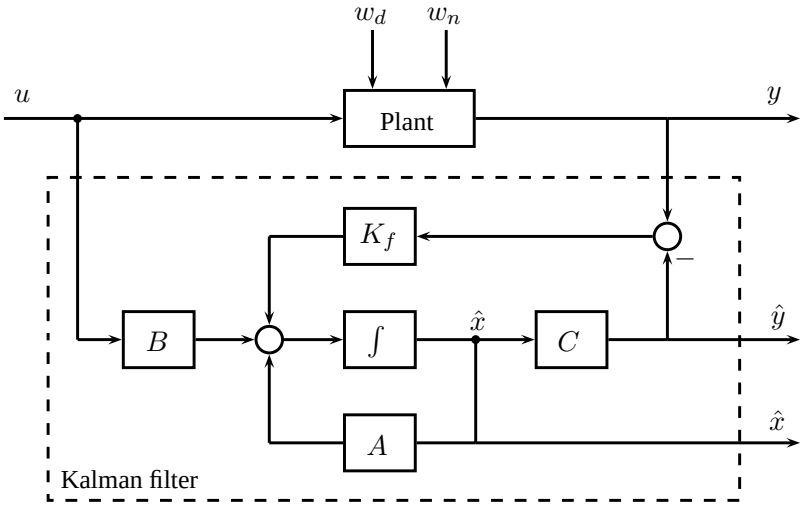


Figure 2.2: The Kalman filter and noisy plant.

2.3.6 The LQG problem

The LQG control problem is to find the optimal control input $u(t)$ which minimizes the cost function

$$J = E\left\{ \min_{T\to\infty} \frac{1}{T} \int_0^T \left[x(t)^\top Q x(t) + u(t)^\top R u(t) \right] dt \right\}, \tag{2.38}$$

where Q and R are constant weighting matrices that satisfy $Q \geq 0$ and $R > 0$.[6] The effect of these matrices have already been discussed in section 2.3.3. The name LQG arises from the use of a Linear model, Quadratic cost function (2.38), and Gaussian white noise processes to model disturbances and noise signals (2.19).

The solution to the LQG problem is known as the *separation theorem* or *certainty equivalence principle*. It is surprisingly simple and consists of first solving the deterministic linear quadratic regulator (LQR) problem: the above problem without w_d and w_n. The solution to the LQR problem is a simple state feedback law

$$u(t) = -K_r x(t), \tag{2.39}$$

where K_r is a constant gain matrix which is easy to compute and independent on the statistical properties of the plant noise, that is W and V. The LQR control law requires that x is measured and available for feedback. This difficulty is overcome by computing a optimal estimate $\hat{x}$ of the state x, so that $E\left\{[x-\hat{x}]^\top [x-\hat{x}]\right\}$ is minimized. The optimal estimate is given by a Kalman filter and is independent of Q and R. The solution to the LQG problem is then found by replacing x with $\hat{x}$, to give $u = -K_r \hat{x}$. The separation theorem is simply that the problem can be cast as two distinct subproblems, one being the LQR problem, the second being the optimal estimator problem.[7] The separation theorem is illustrated in Figure 2.3.

[6] For the rest of Chapter 2.3 we only consider continuous systems on the form (2.18) with noise weights (2.19). The presented theory applies with little modification also to discrete systems.

[7] The author likes to think of the separation theorem in terms of *dynamic programming* and the *principle of optimality*, where the optimal substructure is split into the Kalman filter from y to $\hat{x}$, and from $\hat{x}$ to u the LQR law. In fact, most optimal control problems can be solved with dynamic programming by analyzing the appropriate *Hamilton-Jacobi-Bellman equation* [Naidu, 2003].

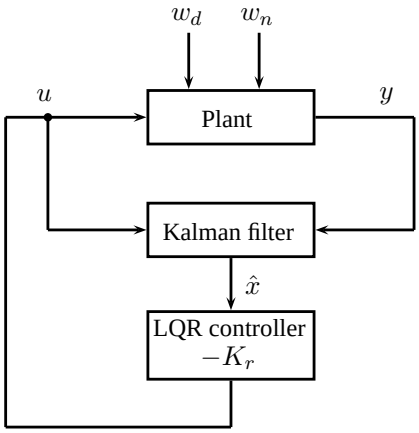


Figure 2.3: Illustration of the separation theorem.

2.3.7 LQG: combined optimal state estimation and optimal state feedback

The combination of an LQR and (steady-state) Kalman filter is a dynamic controller. The transfer function from y to u is (assuming negative feedback)

$$K_{LQG}(s) \stackrel{s}{=} \left[\begin{array}{c|c} A-BK_r-K_fC & K_f \\ \hline -K_r & 0 \end{array}\right]. \tag{2.40}$$

The controller has the same degree (number of poles) as the plant. Using Eqs. (2.34), (2.35), and (2.22) we obtain the closed-loop dynamics

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix} x \\ x-\hat{x} \end{bmatrix} = \begin{bmatrix} A-BK_r & BK_r \\ 0 & A-K_fC \end{bmatrix}\begin{bmatrix} x \\ x-\hat{x} \end{bmatrix} + \begin{bmatrix} I & 0 \\ I & -K_f \end{bmatrix}\begin{bmatrix} w_d \\ w_n \end{bmatrix}. \tag{2.41}$$

The closed-loop poles are simply the union of LQR and kalman filter, as is expected from the separation theorem.

2.3.8 LQG with integral action

The standard LQG design procedure does not have integral action in the controller. This section presents one way of including integral action with LQG. The control structure, depicted in Figure 2.4, is borrowed from [Skogestad & Postlethwaite, 2005].

With this setup the LQG controller stabilizes the plant, and the extra integral effect is used to follow a reference signal. The controller dynamics are

$$\frac{\mathrm{d}}{\mathrm{dt}}\begin{bmatrix}\hat{x}\\ \sigma\end{bmatrix} = \begin{bmatrix}A - BK_{rx} - K_f C & -BK_{ri}\\ 0 & 0\end{bmatrix}\begin{bmatrix}\hat{x}\\ \sigma\end{bmatrix} + \begin{bmatrix}K_f\\ I\end{bmatrix} y, \tag{2.42a}$$

$$u = [-K_{rx} \; -K_{ri}]\begin{bmatrix}\hat{x}\\ \sigma\end{bmatrix}, \tag{2.42b}$$

where we have split the LQR gain into two parts, K_{rx} for the estimated states, and K_{ri} for the integrated outputs σ.

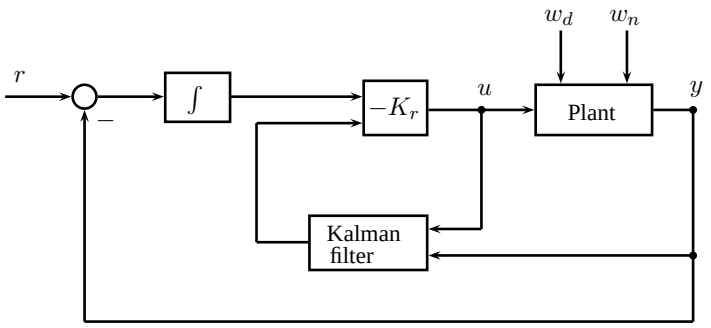


Figure 2.4: LQG with integral action.

2.3.9 $\mathscr{H}_2$-optimal control and link to LQG

The standard $\mathscr{H}_2$ optimal control problem is to find a stabilizing controller K which minimizes

$$||F(s)||_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{tr}\left[F(j\omega)F(j\omega)^H\right] d\,\omega}, \quad F \triangleq F_l(P,K), \tag{2.43}$$

with $F_l(P,K)$ denoting the lower LFT of the generalized plant P and controller K.[8] Note that F must be strictly proper, otherwise the $\mathscr{H}_2$ norm is infinite. The generalized plant P typically includes the plant model, the interconnection structure, and the designer-specific weighting functions. The $\mathscr{H}_2$ norm have several deterministic and stochastic interpretations. A deterministic interpretation of the $\mathscr{H}_2$ norm is the 2-norm of the impulse response, i.e. the output resulting from applying unit impulses to each input (allowing the output to settle to zero before the

[8] The generalized plant and Linear fractional transforms (LFTs) are covered in appendices A.1 and A.2.

applying an impulse to the next input). In the stochastic case, minimizing the $\mathscr{H}_2$ norm equals the minimization of the output power of the generalized system, due to a unit intesity white noise input. That is, when

$$E\left\{w(t)\,w(t)^\top\right\} = I\delta(t-\tau) \tag{2.44}$$

the expected power in the error signal z is

$$E\left\{\lim_{T\to\infty}\frac{1}{2T}\int_{-T}^{T} z(t)^\top z(t)\,dt\right\} = ||F_l(P,K)||_2^2, \tag{2.45}$$

and it is clear that minimizing $\mathscr{H}_2$ norm is the same as minimizing the root-mean-square (rms) value of z. These relations can be shown by using Parseval's theorem.

The LQG problem described in 2.3.6 is a special case of $\mathscr{H}_2$-optimal control. Following the derivation in [Skogestad & Postlethwaite, 2005] we show how the LQG problem can be cast as an $\mathscr{H}_2$ optimization in the general framework. By defining

$$z = \begin{bmatrix} Q^{\frac{1}{2}} & 0 \\ 0 & R^{\frac{1}{2}} \end{bmatrix}\begin{bmatrix} x \\ u \end{bmatrix} \tag{2.46}$$

and representing the stochastic inputs w_d and w_n as

$$\begin{bmatrix} w_d \\ w_n \end{bmatrix} = \begin{bmatrix} W^{\frac{1}{2}} & 0 \\ 0 & V^{\frac{1}{2}} \end{bmatrix} w, \tag{2.47}$$

where w is a white noise process of unit intensity, we can write the LQG cost function as

$$J = E\left\{\lim_{T\to\infty}\frac{1}{T}\int_{0}^{T} z(t)^\top z(t)\,dt\right\} = ||F_l(P,K)||_2^2. \tag{2.48}$$

In this formulation

$$z(s) = F_l(P,K)\,w(s) \tag{2.49}$$

and the generalized plant P is given by

$$P \overset{s}{=} \left[\begin{array}{c|ccc} A & W^{\frac{1}{2}} & 0 & B \\ \hline Q^{\frac{1}{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & R^{\frac{1}{2}} \\ C & 0 & V^{\frac{1}{2}} & 0 \end{array}\right]. \tag{2.50}$$

The above formulation of the LQG problem is illustrated in Figure 2.5.

When the system is deterministic, that is $w_d = 0$ and $w_n = 0$, the LQG problem reduces to the LQR problem. We can formulate the LQR problem in the general framework by defining $w = x_0$, $v = y$ and

$$\begin{bmatrix} z \\ y \end{bmatrix} = P \begin{bmatrix} x_0 \\ u \end{bmatrix}, \tag{2.51}$$

where the generalized plant P is given by

$$P \stackrel{s}{=} \left[\begin{array}{c|cc} A & I & B \\ \hline Q^{\frac{1}{2}} & 0 & 0 \\ 0 & 0 & R^{\frac{1}{2}} \\ C & 0 & 0 \end{array}\right]. \tag{2.52}$$

The relation between z and x_0 can now be written

$$z(s) = F_l(P,K)x_0(s) \tag{2.53}$$

and the effect of x_0 on z can be minimized by minimizing

$$J = ||F_l(P,K)||_2. \tag{2.54}$$

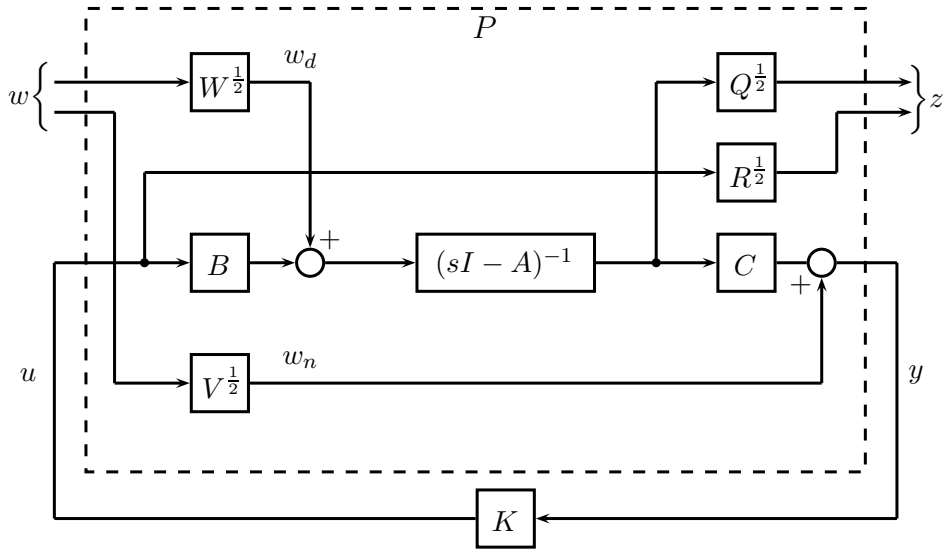


Figure 2.5: The LQG problem formulated in the general control configuration.

2.4 Self-optimizing Control

In *self-optimizing control* (SOC) we look for which variables to control. The objective is to find the optimal combination of measurements which, when controlled to constant setpoint values, will keep the process close to optimum operating conditions despite disturbances and measurement error. The use of a constant setpoint policy can reduce or even eliminate the need for a real-time optimization (RTO) layer (which compute new optimal setpoints). This is one of the major motivations behind self-optimizing control, defined as:

> Self-optimizing control is when we can achieve an acceptable loss with constant setpoint values for the controlled variables without the need to reoptimize when disturbances occur. [Skogestad & Postlethwaite, 2005]

Acceptable loss is quantified in terms of a cost function J which we attempt to minimize. The constant setpoint policy (c_s constant) should yield acceptable loss, $L = J(u,d) - J^{\text{opt}}(d)$, in spite the presence of uncertainty. Uncertainty enters the problem through (i) external disturbances d and (ii) implementation errors $n \triangleq c_s - c$. The implementation error n is caused by (i) steady-state control error n^c and (ii) the measurement error n^y. For linear measurement combinations we thus have $n = n^c + Hn^y$. The control error n^c is determined by the controller, but if we assume that the controller has integral action the steady-state error can be neglected, i.e. $n^c = 0$. The implementation error n is then given by the measurement error, i.e. $n = Hn^y$. Note that it is not possible to have zero loss when implementation errors are present because each new measurement adds a disturbance through its associated measurement error, n^y.

2.4.1 Results from self-optimizing control

We here state the most important results from self-optimizing control. The notation is adopted from [Alstad et al., 2008] and presented in Table 2.3.

The objective is to achieve optimal steady-state operation, where the degrees of freedom u are selected such that the scalar cost function $J(u,d)$ is minimized for any given disturbance d [Skogestad et al., 2003]. It is assumed that any optimally active constraint have been implemented, and that u contains the remaining de-

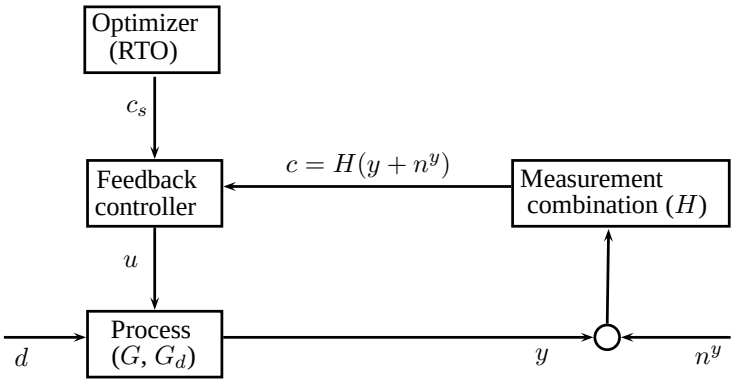


Figure 2.6: Feedback implementation of optimal operation with separate layers for optimization (RTO) and control.

Table 2.3: Notation

u	Vector of n_u unconstrained variables (degrees of freedom)
d	Vector of n_d disturbances
y	Vector of n_y selected measurements used in forming c
c	Vector of selected controlled variables (to be identified) with dimension $n_c = n_u$
n^y	Measurement error associated with y
n^c	Control error associated with c
n	Implementation error associated with c; $n = n^c + Hn^y$

grees of freedom.[9] The reduced optimization problem is

$$\min_u J(u,d). \tag{2.55}$$

A set of controlled variables c which under a constant setpoint policy (where u is adjusted to keep c constant, i.e. $c = c_s$) yields optimal operation, at least locally, is searched for. The problem structure is illustrated in Figure 2.6.

By solving (2.55) for a given d we obtain $J^{\text{opt}}(d)$, $u^{\text{opt}}(d)$, and $y^{\text{opt}}(d)$. Because of changing disturbances and implementation errors $u = u^{\text{opt}}(d)$ can not be implemented in practice. The resulting loss (L) is defined as the difference between the cost J, when using a non-optimal input u, and $J^{\text{opt}}(d)$:

$$L = J(u,d) - J^{\text{opt}}(d). \tag{2.56}$$

[9] See [Skogestad & Postlethwaite, 2005] for a more detailed discussion about the underlying assumptions in self-optimizing control.

A local second-order accurate Taylor series expansion of the cost function around the nominal point (u^*, d^*) is then

$$J(u,d) = J(u^*,d^*) + \begin{bmatrix} J_u & J_d \end{bmatrix}^\top \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix}^\top \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^\top & J_{dd} \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta d \end{bmatrix}, \tag{2.57}$$

where $\Delta u = u - u^*$ and $\Delta d = d - d^*$.[10] For a given disturbance ($\Delta d = 0$), the second-order accurate expansion of the loss function around the optimum ($J_u = 0$) becomes

$$L = \frac{1}{2}(u - u^{\text{opt}})^\top J_{uu}(u - u^{\text{opt}}) = \frac{1}{2} z^\top z, \tag{2.58}$$

where

$$z \triangleq J_{uu}^{1/2}(u - u^{\text{opt}}). \tag{2.59}$$

From here on we consider a constant setpoint policy where the controlled variables are linear combinations of the measurements, i.e.

$$c = Hy. \tag{2.60}$$

It is assumed that the number of available degrees of freedom is equal to the number of controlled variables, i.e. $n_c = n_u$. The constant setpoint policy implies that u is adjusted to give $c_s = c + n$ where n is the implementation error for c. And n is caused by the measurement error, $n = Hn^y$. To find the optimal measurement combination we need to express the loss variables z in terms of d and n^y.

The linearized (local) steady-state model is written as

$$y = G^y u + G_d^y d = \tilde{G}^y \begin{bmatrix} u \\ d \end{bmatrix}, \tag{2.61}$$

$$c = Gu + G_d d, \tag{2.62}$$

where

$$\tilde{G}^y = \begin{bmatrix} G^y & G_d^y \end{bmatrix} \tag{2.63}$$

is the augmented plant. From Eqs. (2.60)–(2.62) we get that

$$G = HG^y \qquad \text{and} \qquad G_d = HG_d^y. \tag{2.64}$$

The disturbances d and measurement errors n^y are quantified by the diagonal scaling matrices W_d and W_n, respectively.

$$d = W_d d', \tag{2.65}$$

$$n^y = W_{n^y} n^{y\prime}, \tag{2.66}$$

[10] The Δ, used to denote deviation variables, is omitted hereafter to simplify the notation.

where we assume that d' and $n^{y'}$ are any vectors satisfying

$$\left\| \begin{bmatrix} d' \\ n^{y'} \end{bmatrix} \right\|_2 \leq 1. \tag{2.67}$$

[Skogestad et al., 2003] justifies the use of the combined vector 2-norm. The nonlinear functions $u^{\text{opt}}(d)$ and $y^{\text{opt}}(d)$ are also linearized, and it can be shown that

$$u^{\text{opt}} = -J_{uu}^{-1} J_{ud} d, \tag{2.68}$$

$$y^{\text{opt}} = -\underbrace{(G^y J_{uu}^{-1} J_{ud} - G_d^y)}_{F} d, \tag{2.69}$$

where F denotes the optimal sensitivity matrix for the measurements. Using the constant setpoint policy and the preceeding equations we derive an expression for the loss variables z in (2.59):

$$z = M_d d' + M_n n^{y'}, \tag{2.70}$$

where

$$M_d = -J_{uu}^{1/2} (HG^y)^{-1} HFW_d, \tag{2.71}$$

$$M_{n^y} = -J_{uu}^{1/2} (HG^y)^{-1} HW_n. \tag{2.72}$$

Introducing

$$M \triangleq [M_d \quad M_{n^y}] \tag{2.73}$$

gives $z = M \begin{bmatrix} d' \\ n^{y'} \end{bmatrix}$, which is the desired expression for the loss variables. From (2.58) we have that a non-zero value for z gives a loss $L = \frac{1}{2}||z||_2$, and the worst-case loss for the expected disturbances and noise in (2.67) is then

$$L_{wc} = \max_{\left\| \begin{bmatrix} d' \\ n^{y'} \end{bmatrix} \right\|_2 \leq 1} L = \frac{1}{2} (\bar{\sigma}[M])^2. \tag{2.74}$$

Thus, to minimize the worst-case loss we need to minimize $\bar{\sigma}(M)$ with respect to H. The problem can be stated as

$$H = \arg\min_H \bar{\sigma}(M). \tag{2.75}$$

This is the "exact local method" in [Skogestad et al., 2003]. At first glance (2.75) may seem like a non-trivial optimization problem as M depends nonlinearly on H, as shown in (2.71)-(2.73). However the optimization problem can be reformulated by introducing

$$M_n \triangleq J_{uu}^{1/2} (HG^y)^{-1} = J_{uu}^{1/2} G^{-1}, \tag{2.76}$$

which may be viewed as the effect of n on the loss variables z. We then have

$$M = [M_d \quad M_{n^y}] = -M_n H[FW_d \quad W_{n^y}]. \tag{2.77}$$

As pointed out in [Alstad et al., 2008] the solution of Eq. (2.75) is not unique. If H is an optimal solution of (2.75), then $H_1 = DH$ is another optimal solution, where D is a non-singular matrix of dimension $n_u \times n_u$. This follows from the fact that M_d and M_{n^y} in (2.71) and (2.72) are unaffected by the choice of D. An implication of this is that M_n in Eq. (2.76) may be selected freely. By introducing the constraint $M_n = I$ and denoting

$$\tilde{F} \triangleq [FW_d \quad W_{n^y}], \tag{2.78}$$

the optimization problem in (2.75) can be stated as

$$H = \arg\min_H \bar{\sigma}(H\tilde{F}) \quad \text{subject to } HG^y = J_{uu}^{1/2}. \tag{2.79}$$

The reformulated problem is easy to solve numerically because of the linearity in H in both the matrix $H\tilde{F}$ and in the equality constraints. [Alstad et al., 2008] provides a discussion on the properties of the optimization problem in (2.79) and gives a discussion of possible choice of norm (Frobenious gives the same result as the singular value or induced 2-norm). An explicit solution of the problem, when $\tilde{F}\tilde{F}^\top$ has full row rank, is proved to be

$$H^\top = (\tilde{F}\tilde{F}^\top)^{-1}G^y(G^{y\top}(\tilde{F}\tilde{F}^\top)^{-1}G^y)^{-1}J_{uu}^{1/2}. \tag{2.80}$$

The results presented in this chapter are summarized in the following theorem from [Alstad et al., 2008].

Theorem 2.3 (Exact local method: Optimal measurement combination)**.** *For combined disturbances and measurement errors, the optimal measurement combination in terms of the Frobenius-norm can be formulated as*

$$\min_H ||H\tilde{F}||_F \quad \textit{subject to } HG^y = J_{uu}^{1/2}, \tag{2.81}$$

where $\tilde{F} = [FW_d \quad W_{n^y}]$. F is the optimal measurement sensitivity with respect to disturbances, and W_d and W_{n^y} are diagonal weighting matrices, giving the magnitudes of the disturbances and measurement noise, respectively. Assuming $\tilde{F}\tilde{F}^\top$ is full rank, the explicit solution for the combination matrix H is

$$H^\top = (\tilde{F}\tilde{F}^\top)^{-1}G^y(G^{y\top}(\tilde{F}\tilde{F}^\top)^{-1}G^y)^{-1}J_{uu}^{1/2}. \tag{2.82}$$

This solution also minimizes the singular value of M, $\bar{\sigma}(M)$, that is, provides the solution to the "exact local method" in (2.75).

Remark. The worst-case loss for the expected disturbances and noise in (2.67), introduced by adding the constraint $c = Hy$, is $L_{wc} = \bar{\sigma}(M)/2$, where M is given by Eqs. (2.71)-(2.73).

Remark. The convex optimization problem in Theorem 2.3 can be solved using for example CVX, a package for specifying and solving convex programs [Grant & Boyd, 2009], with the following code:

```
cvx_begin
  variable H(nu*N,ny+nu*N);
  minimize norm(H*Ftilde, 'fro')
  subject to
    H*Gy == sqrtm(Juu);
evx_end
```

Remark. The above derivations are local, since we assume a linear process and a second-order objective function in the inputs and the disturbances. Thus, the proposed controlled variables are only globally optimal for the case with a linear model and a quadratic objective.

Remark. The results from Theorem 2.3 can be interpreted as adding linear constraints that minimize the effect on the solution to a quadratic optimization problem.

2.4.2 Nullspace method

The nullspace method is a simple method that give the optimal measurement combination H. The method is based on the following observation: by neglecting the implementation error ($M_{n^y} = 0$), we obtain zero loss ($M_d = 0$) in (2.71) by selecting H such as $HF = 0$ ([Alstad & Skogestad, 2007] & [Alstad et al., 2008]). The nullspace method is summarized in the below theorem.

Theorem 2.4 (Nullspace method)**.** *Consider an unconstrained quadratic optimization problem in the variables u (vector of n_u unconstrained, independent, and free variables) and d (vector of n_d independent disturbances)*

$$\min_u J(u,d) = \begin{bmatrix} u & d \end{bmatrix} \begin{bmatrix} J_{uu} & J_{ud} \\ J_{ud}^\top & J_{dd} \end{bmatrix} \begin{bmatrix} u \\ d \end{bmatrix}. \tag{2.83}$$

Assume that we want to obtain $n_c = n_u$ independent controlled variables c that are linear combinations of the measurements

$$c = Hy. \tag{2.84}$$

Let $y = G^y u + G_d^y d$ be the candidate measurements and $F = \frac{\partial y^{opt}}{\partial d^\top} = -(G^y J_{uu}^{-1} J_{ud} - G_d^y)$ the optimal sensitivity matrix evaluated with constant active constraints. If

there exist $n_y \geq n_u + n_d$ independent measurement, it is possible to select the matrix H in the left nullspace of F, $H \in N(F^\top)$, such that we get

$$HF = 0. \tag{2.85}$$

With this choice for H, fixing c (at its nominal optimal value) is first-order optimal for disturbances d; that is, the loss is zero as long as the sensitivity matrix F does not change.

Remark. Requiring $n_y \geq n_u + n_d$ independent measurements is the same as requiring that the matrix $\tilde{G}^y = [G^y \quad G_d^y]$ is of rank larger than or equal to n_y, assuming that the vectors u and d consist of independent variables.

Remark. The main disadvantage with the nullspace method is that we have no control of the loss caused by measurement errors as given by the matrix $M_{n^y} = -M_n H W_{n^y}$. In [Alstad et al., 2008] they derive an explicit expression for H which is used to calculate M_{n^y}, and to extend the nullspace method to cases with extra or too few measurements, i.e. $n_y \neq n_u + n_d$. This is the *Extended nullspace method* proposed in the same paper.

Chapter 3

Review of literature

In this chapter we review the recent results from the research on *convex initialization of the $\mathscr{H}_2$-optimal static output feedback problem* [Manum et al., 2009]. The theory from Chapter 2 is used throughout this review and the reader is advised to consult it whenever needed.

3.1 Introducing the problem

In [Manum et al., 2007] a link between invariants for quadratic optimization problems and linear quadratic (LQ) optimal control was established. The link is that for LQ control one invariant is $c_k = u_k + Kx_k$, which yields zero loss from optimality when controlled to a constant setpoint $c = c_s = 0$. The idea presented in [Manum et al., 2009] is to use this link to generate a controller K_0 suitable for initializing a numerical search for the $\mathscr{H}_2$-optimal static output feedback.

Consider a finite horizon LQ problem on the form:

$$\min_u J(u, x(0)) = E\left\{ x_N^\top P x_N + \sum_{k=0}^{N-1} x_k^\top Q x_k + u_k^\top R u_k \right\} \tag{3.1a}$$

$$\text{subject to } x_{k+1} = Ax_k + Bu_k, \tag{3.1b}$$

$$y_k = Cx_k + w_n, \tag{3.1c}$$

$$x_0 = x(0), \tag{3.1d}$$

where $x_k \in \mathbb{R}^{n_x}$ are the states, $u_k \in \mathbb{R}^{n_u}$ are the inputs, and $y_k \in \mathbb{R}^{n_y}$ are the outputs. The inputs are collected in the vector $u = (u_0, u_1, \ldots, u_{N-1})$. As usual $Q \geq 0$, $R > 0$, and $P = P^\top \geq 0$ are matrices of appropriate dimensions. $E\{\cdot\}$ is the expectation operator.

When $C = I$ and $w_n = 0$ the problem reduces to the well-known LQR problem and the solution to (3.1) is the state feedback $u_k = -Kx_k$. If w_n and x_0 follow the white noise assumption, the solution to (3.1) is the feedback law $u_k = -K\hat{x}_k$, where $\hat{x}_k$ is the state estimate from a Kalman filter. The resulting LQG controller is then of same order n_x as the plant.

In [Manum et al., 2009] they considers the static output feedback problem, $u_k = K^y x_k$, using the LQ setting above. In particular they consider the multi-input multi-output (MIMO) PID controller with the number of controlled output n_c equal to the number of inputs n_u. They allow measurements y_k on the form of controlled outputs y_k^c (P), the integrated value $\sum_{i=0}^{k} y_k^c$ (I), and the derivative $\frac{\partial y_k}{\partial t}$ (D). Using results from self-optimizing control [Alstad et al., 2008] they propose a convex approach to find an initial estimate of K^y, suitable as a starting point for a numerical search. The approach is repeated in the next section and the main algorithm is presented at the end of this chapter.

3.2 Finding the optimal measurement combination H

Here we consider a few important cases where we can find explicit expressions for the problem specific matrices: G^y, G_d^y, J_{uu}, J_{ud}, F, and H.

Case 1) Full information: In this case we assume that noise-free measurements of all states are available. The LQ problem in (3.1) can then be rewritten to the form in (2.83) by treating x_0 as the disturbance, and letting $u = [u_0, u_1, \ldots, u_{N-1}]^\top \in \mathbb{R}^{\tilde{n}_u}$. From Theorem 2.4 we know that there exists infinitely many invariants (but only one of these involves only present states).

Let the measurement candidates be $y_c = [x_0, u_0, u_1, \ldots, u_{N-1}]^\top = [x_0, u]^\top \in \mathbb{R}^{\tilde{n}_y}$. Note that this also includes future inputs, however, we will use the usual trick from MPC and only implement the present (first) input u_0. Since we have that $\tilde{n}_y =$

$\tilde{n}_u + n_d$ we can utilize Theorem 2.4 with the open-loop model:

$$y_c = G^y u + G_d^y d, \tag{3.2}$$

$$G^y = \begin{bmatrix} 0_{n_x \times n_u N} \\ I_{n_u N} \end{bmatrix} \in \mathbb{R}^{(n_x + n_u N) \times (n_u N)}, \tag{3.3}$$

$$G_d^y = \begin{bmatrix} I_{n_x} \\ 0_{n_u N \times n_x} \end{bmatrix} \in \mathbb{R}^{(n_x + n_u N) \times (n_x)}, \tag{3.4}$$

where we have introduced the notation I_m for an identity matrix with dimension $m \times m$.

The matrices J_{uu} and J_{ud} are derivatives of the linear quadratic objective function. They are derived in B.2 and shown to be:

$$\frac{J_{uu}}{2} = \begin{bmatrix} B^\top PB + R & B^\top A^\top PB & \dots & B^\top (A^{N-1})^\top PB \\ B^\top PAB & B^\top PB + R & \dots & B^\top (A^{N-2})^\top PB \\ \vdots & \vdots & \ddots & \vdots \\ B^\top PA^{N-1}B & B^\top PA^{N-2}B & \dots & B^\top PB + R \end{bmatrix}, \tag{3.5}$$

$$\frac{J_{ud}}{2} = \begin{bmatrix} B^\top P \\ B^\top PA \\ \vdots \\ B^\top PA^{N-1} \end{bmatrix} A. \tag{3.6}$$

In this case the sensitivity matrix becomes:

$$F = -(G^y J_{uu}^{-1} J_{ud} - G_d^y) = \begin{bmatrix} I_{n_x} \\ -J_{uu}^{-1} J_{ud} \end{bmatrix}. \tag{3.7}$$

Using Theorem 2.4 we have that the combination matrix H is given by $HF = 0$. Partitioning H gives

$$[H_1 \quad H_2] \begin{bmatrix} I_{n_x} \\ J_{uu}^{-1} J_{ud} \end{bmatrix} = H_1 - H_2 (J_{uu}^{-1} J_{ud}) = 0. \tag{3.8}$$

To ensure a non-trivial solution we can choose $H_2 = I_{n_u N}$. The optimal combination of x_0 and u is then

$$c = Hy = \underbrace{J_{uu}^{-1} J_{ud}}_{K} x_0 + u \Rightarrow u = -Kx_0, \tag{3.9}$$

which can be interpreted as:

$$\begin{aligned} &\text{Invariant 1: } u_0 = -K_0 x_0 \\ &\text{Invariant 2: } u_1 = -K_1 x_0 \\ &\qquad \vdots \\ &\text{Invariant N: } u_{N-1} = -K_{N-1} x_0 \end{aligned} \tag{3.10}$$

From Theorem 2.4 implementation of (3.10) give zero loss from optimality, i.e. they correspond to the optimal trajectory $u_0^*, u_1^*, \ldots, u_{N-1}^*$ from the solution of (3.1). Moreover, since the states capture all information, we must have that

$$u_1 = -\underbrace{K_1(A-BK_0)^{-1}}_{K_0} x_1. \tag{3.11}$$

From this we deduce that the solution of (3.1) can be implemented as $u_k = -K_0 x_k$. In [Manum et al., 2007] it was proved that this gives the same result as conventional linear quadratic control (LQR).

Case 2) Output feedback: We now consider the static output feedback case where $u_k = -K^y y_k$ and $y_k = Cx_k + 0 \cdot u_k$, $k = 0, 1, \ldots, N$. When C has full column rank we have full information (state feedback), but we here consider the general case where C has full row rank (independent measurements), but not full column rank.

Let $y_c = [y_0, u]^\top \in \mathbb{R}^{\tilde{n}_y}$, as before $u = [u_0, u_1, \ldots, u_{N-1}] \in \mathbb{R}^{\tilde{n}_u}$ and the disturbance is $d = x_0$. The open-loop model is now

$$y_c = \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{G^y} u + \underbrace{\begin{bmatrix} C \\ 0 \end{bmatrix}}_{G_d^y} d, \tag{3.12}$$

and the sensitivity matrix F becomes

$$F = -(G^y J_{uu}^{-1} J_{ud} - G_d^y) = \begin{bmatrix} C \\ -J_{uu}^{-1} J_{ud} \end{bmatrix}. \tag{3.13}$$

We then have that $\tilde{n}_y = n_y + \tilde{n}_u \leq n_d + \tilde{n}_u$ and we cannot simply set $HF = 0$, but we need to solve the optimization problem in Theorem 2.3.

We follow [Manum et al., 2009] and try to analyze the problem for a given measurement combination H. Lets start by splitting H in the following manner

$$H_{n_c \times (n_y + \tilde{n}_u)} = [H_{1\, n_c \times n_y} \quad H_{2\, n_c \times \tilde{n}_u}]. \tag{3.14}$$

For $G^y = [0 \quad I]^\top$, $HG^y = J_{uu}^{1/2}$ is equivalent to $H_2 = J_{uu}^{1/2}$. Further we have that

$$H\tilde{F} = H[FW_d \quad W_{n^y}] = [HFW_d \quad HW_{n^y}], \tag{3.15}$$

and for the noise-free case where $W_{n^y} = 0$ we get

$$H\tilde{F} = [HFW_d \quad 0]. \tag{3.16}$$

The objective is to minimize the Frobenius norm of this matrix, that is we want to minimize

$$||[HFW_d \quad 0]||_F = ||HFW_d||_F + ||0||_F. \tag{3.17}$$

Assume without loss of generality that $W_d = I$, and let $J = -J_{uu}^{-1}J_{ud}$. Using Eq. (3.13) we get $F^\top = [C^\top \quad J^\top]$ and

$$HF = H_1C + H_2J = H_1C - J_{uu}^{-1/2}J_{ud}, \tag{3.18}$$

where we have used that $H_2 = J_{uu}^{1/2}$. Hence, we want to minimize $||H_1C - J_{uu}^{-1/2}J_{ud}||_F$ and we look for a H_1 so that

$$H_1C = J_{uu}^{-1/2}J_{ud}. \tag{3.19}$$

We can solve this equation for H_1 by using the Moore-Penrose pseudo-inverse, here denoted with a dagger:

$$H_1 = J_{uu}^{-1/2}J_{ud}C^\dagger. \tag{3.20}$$

The optimal measurement combination H is then

$$H = [J_{uu}^{-1/2}J_{ud}C^\dagger \quad J_{uu}^{1/2}]. \tag{3.21}$$

In the final implementation we can decouple the inputs by premultiplying H with $J_{uu}^{-1/2}$, i.e.

$$\tilde{H} = J_{uu}^{-1/2}H = [-J_{uu}^{-1}J_{ud}C^\dagger \quad I]. \tag{3.22}$$

This means that the open-loop optimal output feedback is

$$u_k = -\underbrace{J_{uu}^{-1}J_{ud}}_{K}C^\dagger y_k, \tag{3.23}$$

where K is the state feedback found in the previous case (with full information). So, for an optimal state feedback K the optimal output feedback is $KC^\dagger$. This means that we can write:

$$\begin{aligned}
&\text{"Invariant" 1: } u_0 = -K_0C^\dagger x_0 \\
&\text{"Invariant" 2: } u_1 = -K_1C^\dagger x_0 \\
&\qquad\vdots \\
&\text{"Invariant" N: } u_{N-1} = -K_{N-1}C^\dagger x_0
\end{aligned} \tag{3.24}$$

The variable combinations in 3.24 are called ”invariants” in quotation marks because they are not invariant to the solution of the original problem, but rather the variable combinations that minimize the (open-loop) loss. Indeed, the non-negative loss is

$$||HF|| = ||J_{uu}^{-1/2}J_{ud}C^\dagger C - J_{uu}^{-1/2}J_{ud}|| \tag{3.25}$$

$$= ||J_{uu}^{-1/2}J_{ud}(C^\dagger C - I)|| \tag{3.26}$$

$$\leq ||J_{uu}^{-1/2}J_{ud}|| \cdot ||(C^\dagger C - I)||. \tag{3.27}$$

For output feedback we have in the least squares sense that

$$u_0 = -\underbrace{K_0 C^\dagger}_{K_0^y} y_0, \tag{3.28}$$

$$u_1 = -\underbrace{K_1 C^\dagger C(A - BK_0 C^\dagger C)^{-1} C^\dagger}_{K_1^y} y_1. \tag{3.29}$$

Unfortunately, in general $K_1^y \neq K_0^y$ and hence the open-loop solution in (3.24) cannot be implemented as a constant feedback $u_k = -K_0 C^\dagger y_k = -K_0^y y_k$, as was the case with state feedback.

Case 3) Noisy measurement of full state vector: The third case we will consider is when noisy measurements of the state vector are available, and the noise-level on all states are the same. That is, $\tilde{y}_k = x_k + \alpha$, with $\tilde{y}_k$ denoting the noisy measurement vector and α the *noise-to-disturbance* ratio. As before, we treat the inital state as a disturbance ($d = x_0$). The disturbance and noise weights become

$$W_d = I,\ W_{n^y} = \alpha I, \tag{3.30}$$

which are assumed to satisfy the bounds in (2.65), (2.66), and (2.67). As proved in [Manum et al., 2009] the optimal feedback in this case becomes

$$u_k = \frac{1}{1+\alpha^2} K x_k, \tag{3.31}$$

where K is the optimal feedback in the noise-free case ($\alpha = 0$). From (3.31) we see that the optimal state feedback is reduced by a factor $1/(1+\alpha^2)$ compared to the noise-free case, and, that the optimal gain decreases as the noise level α goes up.

3.3 Closed-loop optimization

In this context we say that a feedback law is *open-loop LQ-optimal* if it minimizes the quadratic objective, subject to (linear) system constraints, with the future inputs $u = [u_0, u_1, \ldots, u_{N-1}]^\top$ as optimization variables. We further call such an optimization for an *open-loop optimization*. The LQR is an example of an open-loop LQ-optimal controller where the optimal sequence of future inputs can be written as $u_k = -K_r x_k$. This property of the LQR controller is desirable as it can be implemented as a simple feedback.

We also introduce the terms *closed-loop LQ-optimal* controller and *closed-loop optimization*, which differs from the open-loop terms in that the optimization problem is solved with respect to the controller K, i.e. by writing

$$\min_K J(K), \tag{3.32}$$

we mean minimize J over the parameters in K. Then, if the optimization problem is convex we can find the globally optimal K which minimizes J and implement it as a static feedback. Unfortunately, it is generally difficult to prove convexity of such closed-loop optimization problems. This has already been touched upon in Chapter 2.1 and it will be the topic of the next chapter in this thesis.

A closed-loop optimization problem can be specified in several ways. In [Manum et al., 2009] they use the $\mathscr{H}_2$ formulation described in Section 2.3.9, that is

$$\min_K ||F_l(P,K)||_2, \tag{3.33}$$

where $||F_l(P,K)||_2$ is the $\mathscr{H}_2$ norm of the lower LFT of the generalized plant P and the controller K.[1] This formulation may be implemented in an LQG-scheme and the open-loop LQ-optimal controller from Section 3.2 can be used to initialize the closed-loop optimization problem.

In the deterministic case, with $V = 0$, $W = I$, and P defined as in Eq. (2.52), the $\mathscr{H}_2$ problem represents the LQR problem. The $\mathscr{H}_2$ norm can then be interpreted as the sum of impulse responses resulting from unit impulses in each input (in this setting x_0 is regarded as the input), and we can use the following approximation of the $\mathscr{H}_2$ problem in (3.33):

$$\min_K \sqrt{\text{trace}(M(K))}. \tag{3.34}$$

The function $M(K)$ is derived in Appendix B.3.

[1] By following the standard notation used in the control community we here get some ambiguous notation. The reader should not confuse the generalized plant with the final state weight in the quadratic objective function, both denoted P. It should be clear from the context what is meant when P occurs in the text.

3.4 Initialization of the $\mathscr{H}_2$-optimal static output feedback problem

A slightly altered version of the main algorithm in [Manum et al., 2009] is presented below.[2] It is a two-step procedure for finding the $\mathscr{H}_2$-optimal static output feedback. The first step (1-6) is to find the open-loop LQ-optimal controller K_0. This controller corresponds to a controller that in the open-loop sense is closest to the optimal state feedback LQ controller. The last step (7) is to improve control by performing a closed-loop optimization using K_0 as the starting value.

Table 3.1: Main algorithm – Low-order controller synthesis

1:	Define a finite-horizon quadratic objective on the form: $J(u,x) = x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i$.
2:	Calculate J_{uu} and J_{ud} as in (3.5) and (3.6).
3:	Define candidate variables $y = G^y u + G_d^y d$, with $u = [u_0, u_1, \ldots, u_{N-1}]^\top$.
4:	Decide disturbance and noise weights W_d and W_{n^y} (Default: $W_d = I$, $W_{n^y} = 0$).
5:	Find H by solving the convex optimization problem in Theorem 2.3.
6:	Deduce the inital LQ-optimal controller K_0 (first invariant) from H.
7:	Improve control by performing a closed-loop optimization with K_0 as the starting value. The closed-loop $\mathscr{H}_2$-optimal controller K can be found by solving either (3.33) or the approximation (3.34).

Remark. The algorithm above is referred to as Algorithm 3.1 from here on.

Remark. With full information (Case 1) we can calculate K_0 directly by extracting the first invariant from $J_{uu}^{-1} J_{ud}$. With output feedback (Case 2) K_0 is the first invariant of $J_{uu}^{-1} J_{ud} C^\dagger$. In both cases we can also find K_0 by performing Step 5 in Algorithm 3.1, i.e. by solving the convex optimization problem in Theorem 2.3.

Remark. In Algorithm 3.1 we have chosen to use the notation from Chapter 2.4 for the disturbance and noise weight, i.e. W_d and W_{n^y} respectively. In Chapter 2.3 we used W and V for the same weights. To avoid confusion we remark here that $W = W_d$ and $V = W_{n^y}$.

[2] The original algorithm is modified by adding Step 6.

Chapter 4

Convexity of the $\mathscr{H}_2$-optimal static output feedback problem

In the previous chapter we reviewed an initialization scheme for the $\mathscr{H}_2$-optimal static output feedback problem, recently proposed in [Manum et al., 2009]. The proposed algorithm is a two-step procedure where the first step finds an initial controller K_0 by solving a convex optimization problem. In the second step K_0 is used to initialize a closed-loop optimization where the controller parameters are the degrees of freedom. The closed-loop optimization is posed in terms of the $\mathscr{H}_2$ norm (3.33) or the impulse response representation (3.34). To remind the reader we restate the closed-loop optimization problem below ($\mathscr{H}_2$ problem):

$$\min_K J_{cl} = ||F_l(P,K)||_2, \tag{4.1}$$

where K is a static controller. If the cost function J_{cl} is convex in K we can find the globally $\mathscr{H}_2$-optimal controller, lets denote it K^*. However, when J_{cl} is non-convex there exist local minima and we can no longer be sure to find K^* using convex programming. In this chapter we investigate the convexity properties of the $\mathscr{H}_2$ problem above. The question we will try to answer is:

- When is the $\mathscr{H}_2$ problem in (4.1) convex?

In other words we want to know when we can find K^* by solving the optimization problem in (4.1) as a convex problem.

In Chapter 2.1 we mentioned that the problem of finding an optimal static output feedback, or the set of all stabilizing static output feedback controllers, is re-

garded as an unsolved problem. All research on the problem indicates that it is $\mathscr{NP}$-hard, meaning that algorithms much more effective than a *brute-force* approach probably does not exist. For example the LQ optimization problem 2.8 results in three coupled, nonlinear matrix equations. These equations are very hard to solve. Other approaches, such as the Lyapunov approach, also give matrix equations which cannot be solved in polynomial time.

The reason that we want to answer the question of convexity is simply that we want to find the optimal static output feedback. It is always preferable (and cheapest) to implement the simplest solution and if a static output feedback yields acceptable control performance there is no doubt in which controller to choose. From [Syrmos et al., 1997] we learned that systems with higher order than the controller can be brought back to the static output feedback problem. Thus, from the thousands of PI and PID loops running in the industry we know that the static output feedback problem is applicable to many real systems.

Preliminary studies have shown that the problem in special cases (SISO systems with one state) can be made convex under certain requirements on the weight matrices Q and R, see Appendix C for details. Before the reader continues, the author wants to underline that no attempt is made here to derive a mathematical proof which answers the question of convexity. Remembering the advice from [Syrmos et al., 1997] we will use the structure of some specific examples to study the convexity of the $\mathscr{H}_2$ problem, hoping that it may lead to new knowledge. But, before presenting these interesting examples a short introduction to convex programming is given.

4.1 Convexity of optimization problems

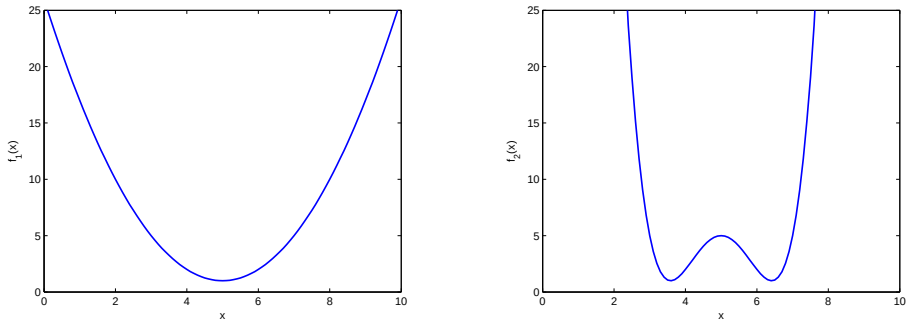
Convexity is a highly attractive property of optimization problems. Problems possessing this property are generally much easier to solve both in theory and in practice.

The term "convex" can be applied both to sets and to functions. Using the definition in [Nocedal & Wright, 2006], we say that a set $S \in \mathbb{R}^n$ is a *comvex set* if the straight line segment connecting any two points in S lies entirely inside S. The function f is a *convex function* if its domain S is a convex set and if for any two points x and y in S, the following property is satisfied:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y), \ \forall \alpha \in [0,1]. \tag{4.2}$$

We say that f is *strictly convex* if the inequality in (4.2) is strict whenever $x \neq y$ and α is in the open interval $(0, 1)$.

Simple instances of convex sets include the unit ball $\{y \in \mathbb{R}^n \mid ||y||_2 \leq 1\}$; and any polyhedron $\{x \in \mathbb{R}^n \mid Ax = b, Cx \leq d\}$. Examples of convex functions are the affine (or linear) function $f(x) = c^\top x + \alpha$, for any constant vector $c \in \mathbb{R}^n$ and scalar α; and the convex quadratic function $f(x) = x^\top Hx$, where H is a symmetric positive semidefinite matrix. Figure 4.1 shows a convex and non-convex function.



(a) A convex function

(b) A non-convex function

Figure 4.1: Example of a convex and non-convex function. The convex function has a global minima, while the non-convex function has to local minima.

Consider the general constrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{array}{l} c_i(x) = 0, i \in \mathscr{E}, \\ c_i \leq 0, i \in \mathscr{I}, \end{array} \tag{4.3}$$

where $\mathscr{E}$ and $\mathscr{I}$ are sets of indicies for equality and inequality constraints, respectively. Then, the term *convex programming* is used to describe the special case of (4.3) in which;

- the objective (cost) function $f(x)$ is convex,
- the equality constraint functions $c_i(\cdot)$, $i \in \mathscr{E}$, are linear, and
- the inequality constraint functions $c_i(\cdot)$, $i \in \mathscr{I}$, are convex.

When solving an optimization problem on the form (4.3) we are looking for a *global minimizer* of f, a point where the function attains its least value. Again, we cite [Nocedal & Wright, 2006] and give the formal definitions:

- A point x^* is a *global minimizer* if $f(x^*) \leq f(x)$, for all x.

- A point x^* is a *local minimizer* it there is a neighborhood $\mathscr{N}$ of x^* such that $f(x^*) \leq f(x)$, for all $x \in \mathscr{N}$.

For the unconstrained optimization problem, (4.3) without inequality constrains (i.e. $\mathscr{I} = \emptyset$), we then know that: *when f is convex, any local minimizer x^* is a global minimizer of f.*

The unconstrained optimization problem with the objective function f in Figure 4.1a is convex, thus if we find the local minimizer we know that it also is the global minimizer of f. If we replace f with the function in Figure 4.1b we no longer have a convex problem. This f has two local minimizers, thus, if we find one of them we are not guaranteed that it is the global minimizer. The other minimizer could be a better solution.

4.2 Examples

To investigate the convexity of the $\mathscr{H}_2$ problem in (4.1) we study a few simple examples. Lets start by defining a LTI "test" system

$$x_{k+1} = Ax_k + Bu_k, \tag{4.4}$$

$$y_k = Cx_k, \tag{4.5}$$

with matrices $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$, and $C \in \mathbb{R}^{n_y \times n_x}$. This system may contain controller dynamics. As we discussed in Chapter 2.1 the problem of finding a controller with order less than or equal to the system may be brought back to the static output feedback problem. By defining design weights Q, R, W, V, and the generalized plant (2.50) we now seek a $\mathscr{H}_2$-optimal static output feedback K.

From the theory in 2.3.9 we know that the system must be strictly proper ($D = 0$) for the $\mathscr{H}_2$ norm to exist. Also, when the system is unstable the $\mathscr{H}_2$ norm is infinite. We immediately understand that we cannot find a static output feedback for unstable systems with $B = 0$ or $C = 0$. Thus, to be able to move the eigenvalues of the unstable modes inside the unit disc we must be require that $C \neq 0$ and $B \neq 0$, leading to the following trivial, but necessary conditions for stabilizability with static output feedback.

Necessary condition 1: A necessary condition for stabilizability of an unstable system with static output feedback is that $C \neq 0$ and $D \neq 0$.

We can relax this condition by requiring that only the unstable modes are stabilizable, leading to the next condition.

Necessary condition 2: A necessary condition for stabilizability of an unstable system with static output feedback is that the pair (A, B) is stabilizable and the pair (A, C) is detectable.

When one of these two conditions is violated there is no use in searching for an $\mathscr{H}_2$-optimal controller using (4.1) (the $\mathscr{H}_2$ norm will be infinite for all controllers K). The above conditions are not new results, but leads to some insight in what to expect when running the optimization problem in (4.1). We continue by presenting two examples of SISO systems which satisfies the above conditions, but which do not give a resulting convex problem.

Example 1. We are interested in finding a SISO system which becomes unstable when the gain is increased, and goes from unstable to stable when the gain becomes high enough. With Bode's stability criterion in mind we create a system which phase crosses -180 degrees two times. Consider the transfer function

$$G(s) = \frac{(T_3 s + 1)(T_4 s + 1)}{s(T_1 s + 1)(T_2 s + 1)}, \tag{4.6}$$

where T_i are time constants. The integrator $1/s$ causes the loop phase to start at -90 degrees at $\omega = 0$. By selecting the time constants T_1 and T_2 we can choose at which frequencies we want an additional phase drop, providing a phase drop of -90 degrees each. We can place the zeros by selecting T_3 and T_4, these gives a phase lift of 90 degrees each. With the values $T_1 = 1.5$, $T_2 = 1$, $T_3 = 0.3$, and $T_4 = 0.1$ we get the frequency response in Figure 4.2. Note that with these values the even PIP (Theorem 2.1) is satisfied, which is a requirement for static output stabilizability.

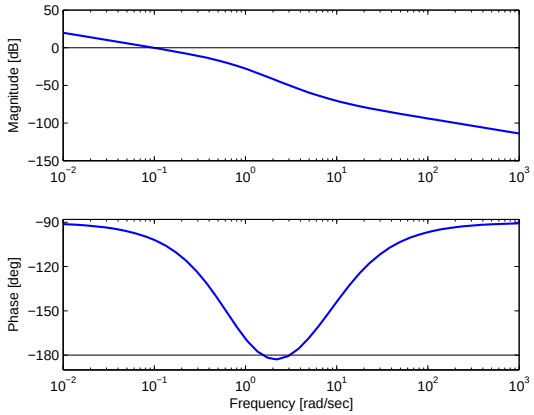


Figure 4.2: Frequency response of $G(s)$.

The controller gain K lifts the amplitude when increased (with negative feedback).

From Bode's stability criterion we know that the system is stable as long as phase is greater than -180 degrees when the amplitude crosses the 0 dB line. Thus, if K is increased too much the 0 dB crossing will occur at the frequencies where the phase is below -180 degrees. However, when K becomes high enough the crossing comes after the phase raises back over -180 degrees, and the system is stable again. For negative K the system is unstable. To summarize we have designed a system which, when K goes from negative to positive and increases towards infinity, has the property that it is: unstable → stable → unstable → stable.

A minimal realization of the system is:

$$A = \begin{bmatrix} -1.6667 & -0.6667 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix}, \tag{4.7}$$

$$C = \begin{bmatrix} 0.0080 & 0.1067 & 0.2667 \end{bmatrix}, \quad D = 0. \tag{4.8}$$

This system is marginally stable with eigenvalues (or poles) at $\{0,\ -0.6667,\ -1\}$. It is both controllable and observable.

To confirm that this system yields a non-convex $\mathscr{H}_2$ problem we plot the objective function in (4.1) ($J_{cl} = ||F_l(P,K)||_2$) as a function of K. The result is presented in Figure 4.3, where we used the weight matrices $Q = 10 \cdot C^\top C$, $R = 0.01$, $W = I$, $V = 0$, and constructed the generalized plant as in (2.50). Note that the $\mathscr{H}_2$ norm is replaced with 50 for values above 50. Most of the replaced values are infinity due to an unstable closed-loop system.

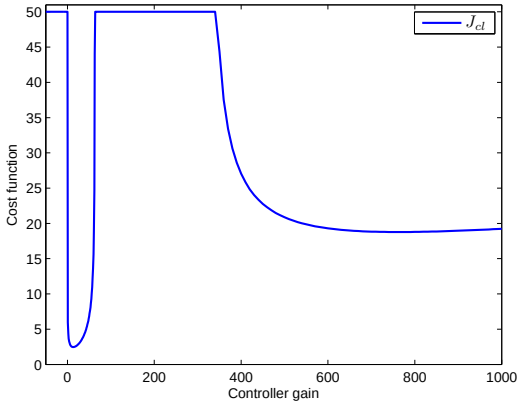


Figure 4.3: The $\mathscr{H}_2$ objective function $J_{cl} = ||F_l(P,K)||_2$ as a function of K.

Example 2. In this example we study a discrete SISO system with matrices:

$$A = \begin{bmatrix} 0.31 & 0.45 & 0.2 \\ 0.33 & 1.11 & 0.1 \\ 0 & -0.51 & 0.15 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.69 \\ 0.38 \end{bmatrix}, \tag{4.9}$$

$$C = \begin{bmatrix} -1 & 0.9 & -0.8 \end{bmatrix}, \quad D = 0. \tag{4.10}$$

This system is one of many found by a *brute-force* algorithm that searches for SISO systems which give a non-convex $\mathscr{H}_2$ problem.[1] The system is controllable, observable, and has eigenvalues at $\{1.1927, 0.0275, 0.3497\}$. With one eigenvalue outside the unit disc the system is unstable. Again we construct the generalized plant using Eq. (2.50), with matrices $Q = I$, $R = 0.01$, $W = I$, $V = 0$. The objective function $J_{cl} = ||F_l(P,K)||_2$ is plottet versus the controller gain K in Figure 4.4 underneath. When the $\mathscr{H}_2$ norm reaches values over 50 we replace it with 50 for the same reasons as in the previous example.

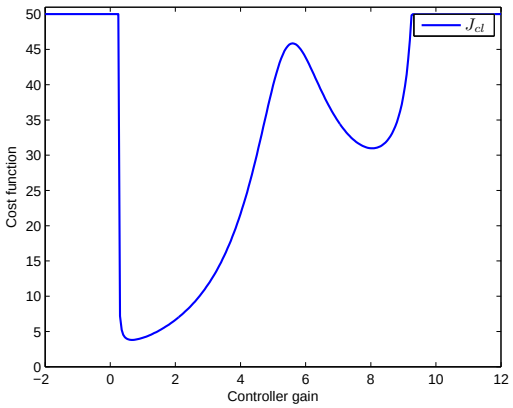


Figure 4.4: The $\mathscr{H}_2$ objective function $J_{cl} = ||F_l(P,K)||_2$ as a function of K.

From Figure 4.4 we identify two local minima for $K = K_1^* = 0.65$ and $K = K_2^* = 8.0$. Clearly, J_{cl} attains its lowest value for $K = K_1^*$ and the global minimum is $J_{cl}^* = 3.81$. Luckily, for this example the initial controller found by Algorithm 3.1 is $K_0 = 3.58$, and the closed-loop optimization converges to the global optimum, J_{cl}^*.

We wish to find the cause for the top between the two minima, when $K \approx 5.8$. A root locus analysis shows that the eigenvalues of the closed-loop system moves toward the boundary of the unit disc when the gain is increased to 5.8, causing a

[1] In more detail, the algorithm changes the parameters of A, B, and C in the search for a $J_{cl} = ||F_l(P,K)||_2$ with more than one minima. On average the algorithm found that about 5% of SISO systems with three states gave a non-convex $\mathscr{H}_2$ problem. This number is based on a test including four thousand different SISO systems.

slower closed-loop response. When K is further increased the eigenvalues fall back in, and the closed-loop response becomes faster, resulting in a local minimum for $K = K_2^*$. If the loop gain becomes too high one of the eigenvalues leaves the unit disc and the closed-loop system is unstable.

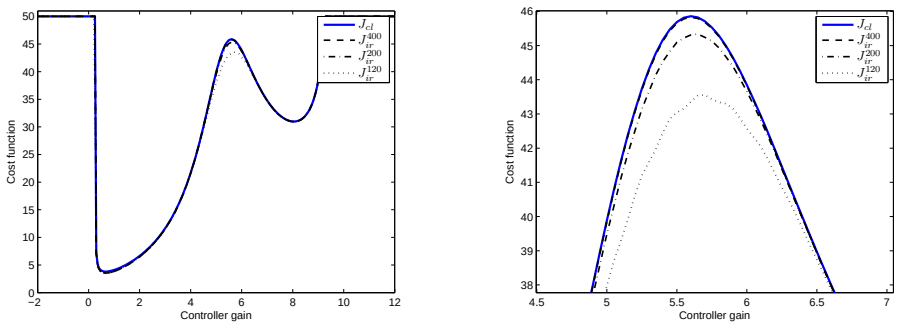
From this short analysis we conclude that SISO systems possessing this fundamental property gives a non-convex $\mathscr{H}_2$ problem. That is, we can only allow the closed-loop response to turn from better to worse one time as the controller gain is increased from zero. Of course, the behaviour of the objective function is is dependant on the choice of weight matrices. For example, if we weigh the states that alternates between slow and fast with zero in the Q matrix, they no longer contribute to the objective function.

Example 3. In this example we study how well the (impulse-response) formulation in (3.34) approximates the $\mathscr{H}_2$ problem in (4.1). From the derivation of (3.34) in Appendix B.3 we know that

$$\min_K J_{ir} = \sqrt{\text{trace}(M(K))} \tag{4.11}$$

is the same as minimizing $||F_l(P,K)||_2$ when $N \to \infty$.

Using the system in *Example 2* we calculate the $\mathscr{H}_2$ norm $J_{cl} = ||F_l(P,K)||_2$ and impulse-response objective function J_{ir} for a range of controller gains. We denote the approximations J_{ir}^{400}, J_{ir}^{200}, and J_{ir}^{120}, where the superscript represents the horizon length (N) utilized. The approximations are compared with J_{cl} in Figure 4.5.



(a) Approximations of the $\mathscr{H}_2$ norm.

(b) A close-up of the figure at the left.

Figure 4.5: Approximation of the $\mathscr{H}_2$ norm for different horizon lengths.

We observe from Figure 4.5 that the approximation improves as N is increased. For $N = 400$ it is hard to distinguish the approximation from J_{cl}.

4.3 Discussion

From Example 1 and 2 we learned that controllability and observability alone (which satisfies necessary condition 1 and 2) is not enough to guarantee a convex $\mathscr{H}_2$ problem. It seems that when the optimization problem is posed in terms of the controller gains we need to elaborate these conditions to include more sophisticated system behaviours. For instance, from Example 2 we know that the set of stabilizing controllers must be a connected set. If it is not, we obtain the undesired behaviour of a system that alternates between being stable and unstable as the controller gains are increased. This results in a non-convex objective function with more than one minima.

The developed *brute-force* algorithm that searches for SISO systems giving a non-convex problem found that about 5 % of the tested systems gave a non-convex problem. This indicates that for a large class of (SISO) systems we can find a globally $\mathscr{H}_2$-optimal static output feedback using Algorithm 3.1. This statement does however require that the first step of the algorithm provides the closed-loop optimization with a feasible starting point, i.e. a stabilizing controller. Furthermore, we saw from Example 2 that the algorithm can be utilized also on non-convex problems, where the solution becomes even more dependant on the initial controller guess. In that specific example the initial guess is good enough for the algorithm to converge to the global optimum.

From Chapter 2.1 we know that in the discrete case there exists a stabilizing gain that minimizes the infinite-horizon LQ problem locally when the system is output stabilizable and C has full row rank (in addition to the common requirements $Q \geq 0$ and $R > 0$). From [Syrmos et al., 1997] we also know that a direct procedure for finding such a controller was unknown at that time. Thus, we are tempted to find out how good the initial controller guess from Algorithm 3.1 is for a more general class of systems. In the next chapter we test the algorithm on the class of stable second-order SISO systems. The goal is to further investigate how the system dynamics affects the convexity of (4.1) and the initial controller guess.

Before moving on we need to comment on the results from Example 3, where we studied how well the (impulse-response) formulation in (3.34) approximates the $\mathscr{H}_2$ problem in (4.1). The formulation in (3.34) is an approximation of the closed-loop infinite-horizon LQ problem (which (4.1) represents) because of the finite horizon length. With (3.34) we assume zero control action after the horizon by choosing the final state weight according to a Lyapunov equation. So, for a given controller we must assume that the states come close enough to zero within the horizon for the approximation to be good. As the controller parameters are the degrees of freedom in the optimization we can not be assured that the states con-

verge fast enough with the different gains, and it becomes hard to choose a satisfying horizon length on beforehand. The solution is to either use a horizon large enough to ensure a good approximation for the most interesting controller gains, or, use the $\mathscr{H}_2$ formulation in (4.1).

Chapter 5

Examples

In this chapter we persue the questions raised in the previous chapter with the aim to learn more about the $\mathscr{H}_2$-optimal static output feedback problem and the workings of Algorithm 3.1. These questions are studied through the three examples introduced next.

We will start with an example where Algorithm 3.1 is utilized on a second-order SISO system. We here show how to synthesize static output feedback (P control), PI and PID control. These controllers are compared against the LQR by studying optimality, stability margins, closed-loop responses, and more. We also study if the initialization procedure is well-suited for the $\mathscr{H}_2$-optimal static output feedback problem in this case.

In the second example we employ the algorithm on a thermal/optical plant with noisy measurements. Noise considerations are included in the synthesis of a MIMO PID and PI controller. These controllers are measured up against the full-order LQG controller by running experiments on the plant. In this elaborate example we wish to see how the controllers tuned by Algorithm 3.1 works in practice.

Finally, we give a basic example of LQG control of a plant that is inherently difficult to control. The point to be made with this example is that some systems are hard to make robust even with a full order controller such as LQG. For such systems it may not be possible to make an static output controller with acceptable performance, even if the system is output stabilizable.

Each example stands on its own feet with an introduction, presentation, and discussion of the obtained results. Therefore, they are gathered in this one chapter.

5.1 $\mathscr{H}_2$-optimal control of a second-order system

In this example we attempt to control a general second-order SISO system using different controllers which are tuned using Algorithm 3.1. We are particularly interested in how well the open-loop controller, found by a convex optimization, initializes the closed-loop optimization in (4.1).

The controllers to be considered are the PID, PD, PI, and P controller. These controllers will be tuned using Algorithm 3.1 and then compared against the LQR controller with both states available. To test the controllers we will use the general second-order system

$$g(s) = \frac{k}{\tau^2 s^2 + 2\tau\zeta s + 1}, \tag{5.1}$$

where ζ is the damping factor. $|\zeta| < 1$ gives an underdamped system with oscillations, while $|\zeta| > 1$ gives an overdamped system. For simplicity we set $k = 1$ and $\tau = 1$. The poles are then located at $s = -\zeta \pm \sqrt{\zeta^2 - 1}$, and we note that the poles move towards the imaginary axis ($\pm j$) as $\zeta \to 0$. The system is realized in state-space form with matrices

$$A = \begin{bmatrix} -2\zeta & -1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D = 0.$$

The system has $n_x = 2$ states, $n_u = 1$ input, and $n_y = 1$ output. (A, B, C, D) is augmented with the respective controller dynamics as in Appendix B.4. The PID controller introduces two new states, the PD and PI controller one, and the P controller zero. We denote the number of new states $\tilde{n}_x$. For PID control the new outputs become the control error $y^p = e$, integrated error $y^i = \int e\,dt$, and derivative approximation of error $y^d \approx \dot{e}$. The derivative is approximated by chosing $\epsilon = 0.1$, see Chapter 2.2. The augmented systems are discretized with a sampling time $T_s = 0.1$. We introduce the notation $(A_{\text{PID}}, B_{\text{PID}}, C_{\text{PID}}, D_{\text{PID}})$ to denote the augmented system, here augmented with the dynamics of a PID controller.

Since the system has two states and only one measurement we have to use Theorem 2.3 to find the optimal measurement combination H. Assuming that all inputs are measured we choose the measurement candidates to be $y_c = [y_0, u]^\top$, where $u = [u_0, u_1, \ldots, u_{N-1}]^\top$ and N is the horizon. y_0 represents the outputs of the augmented system. With $y_c = G^y u + G^y_d d$ and $d = x_0$ we thus get

$$G^y = \begin{bmatrix} D_{\text{PID}} & 0_{n_u \times n_u(N-1)} \\ I_{n_u} & 0 \\ 0 & I_{n_u \times (N-1)} \end{bmatrix} u, \quad G^y_d = \begin{bmatrix} C_{\text{PID}} \\ 0_{n_u \times (n_x + \tilde{n}_x)} \end{bmatrix} d \tag{5.2}$$

where C_{PID} and D_{PID} are replaced with C_{PI} and D_{PI} for PI control, etc.

Following the procedure in Algorithm 3.1 we start by approximating the infinite-horizon quadratic objective with

$$J(x,u) = x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i, \tag{5.3}$$

and the weight matrices in Table 5.1 below. Note that we have chosen to weigh the outputs y^p, y^i, and y^d, instead of the states.

Table 5.1: Weight matrices

	Q	R	P	W	V
LQR	$C^\top C$	1	$C^\top C$	I_{n_x}	0_{n_y}
PID	$C_{\text{PID}}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 0 \end{bmatrix} C_{\text{PID}}$	1	$C_{\text{PID}}^\top \begin{bmatrix} 1 & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 0 \end{bmatrix} C_{\text{PID}}$	$\begin{bmatrix} I_{n_x} & 0 \\ 0 & 0_{2n_y} \end{bmatrix}$	0_{3n_y}
PD	$C_{\text{PD}}^\top \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} C_{\text{PD}}$	1	$C_{\text{PD}}^\top \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} C_{\text{PD}}$	$\begin{bmatrix} I_{n_x} & 0 \\ 0 & 0_{n_y} \end{bmatrix}$	0_{2n_y}
PI	$C_{\text{PI}}^\top \begin{bmatrix} 1 & 0 \\ 0 & 10^{-3} \end{bmatrix} C_{\text{PI}}$	1	$C_{\text{PI}}^\top \begin{bmatrix} 1 & 0 \\ 0 & 10^{-3} \end{bmatrix} C_{\text{PI}}$	$\begin{bmatrix} I_{n_x} & 0 \\ 0 & 0_{n_y} \end{bmatrix}$	0_{n_y}
P	$C^\top C$	1	$C^\top C$	I_{n_x}	0_{n_y}

As seen from Table 5.1 the derivative action y^d is given zero weight. This allows for as much derivative action as needed. The integrator states introduced with PID and PI control gives one pole (eigenvalue) at zero (or unit disc in the discrete case), which unless moved causes the closed-loop $\mathscr{H}_2$ norm to become infinity. Hence, we choose to give the integrated output a small weight to push it inside the left half-plane (or equivalently inside the unit disc). We do not consider noise and set V to zero in all cases. W is chosen to weight disturbances only the original system states, which makes it easier to compare the different controllers in terms of the $\mathscr{H}_2$ norm. Figure 2.5 illustrates how the different weight matrices enter the problem. J_{uu} and J_{ud} are found using Eqs. (3.5) and (3.6), respectively.

We now run the last steps in Algorithm 3.1 to find controllers on the form $u = K_p y^p + K_i y^i + K_d y^d$ (PID), where y^i and y^d are set to zero depending on the desired controller. For instance, $y^i = 0$ gives the PD controller $u = K_p y^p + K_d y^d$. We call the first invariant from the optimal measurement combination H for "first-move" controller and denote it K_0 (initial controller guess). The first-move controller is improved by a closed-loop optimization and we denote the $\mathscr{H}_2$-optimal controller K^*.

From classical control theory we know that integral action lowers the phase and hence the obtainable bandwidth, while derivative action lifts the phase and in-

creases the obtainable bandwidth. Hence, we expect that $J^*_{\text{PI}} \geq J^*_{\text{P}} \geq J^*_{\text{PID}} \geq J^*_{\text{PD}} \geq J^*_{\text{LQR}}$, where $J^* = ||F_l(P,K^*)||_2$ is the $\mathscr{H}_2$ norm of K^*. It is well-known that when $\zeta = 0$ the system cannot be stabilized by a P or PI controller. This is most easily seen from the eigenvalues ($\lambda_i = \pm\sqrt{1+Ki}$) which moves along the imaginary axis for gains $K => -1$ and as $K < -1$ one eigenvalue becomes positive. Thus we expect numerical problems as $\zeta \to 0$ and the algorithm tries to tune the P or PI controller.

Below we present the controller gains for $\zeta = 0.8$, $\zeta = 0.5$, and $\zeta = 0.2$. The results from the convex optimizations are represented by the worst-case loss $L_{\text{wc}} = \bar{\sigma}(M)/2$. We could alternatively have used the open-loop objective function value $J_{ol} = ||H\tilde{F}||_F$ as both rank the controllers equal.

Table 5.2: Controllers for $\zeta = 0.8$

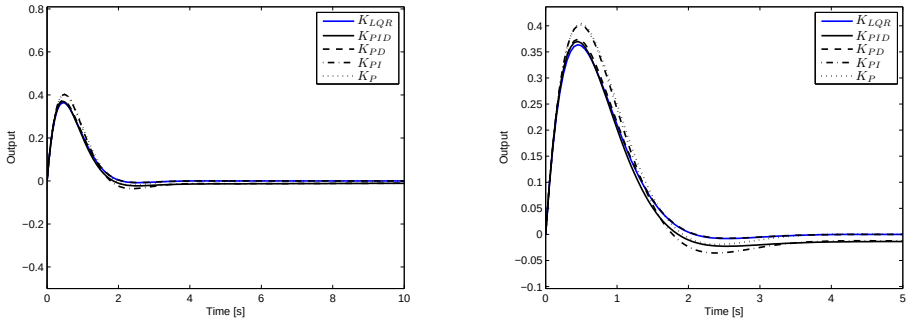
Control law	L_{wc}	$\|\|F_l(P,K^*)\|\|_2$	Controller
$u_k = -0.4050y^p_k - 0.0060y^i_k - 0y^d_k$	0.7555	3.9535	First-move PID
$u_k = -0.4169y^p_k - 0.0310y^i_k - 0.2433y^d_k$	–	**3.7423**	PID
$u_k = -0.3973y^p_k - 0y^d_k$	0.7171	3.6627	First-move PD
$u_k = -0.3787y^p_k - 0.2229y^d_k$	–	**3.6066**	PD
$u_k = -0.4050y^p_k - 0.0060y^i_k$	0.7555	3.9535	First-move PI
$u_k = -0.3247y^p_k - 0.0255y^i_k$	–	**3.7850**	PI
$u_k = -0.3973y^p_k$	0.7171	3.6627	First-move P
$u_k = -0.2945y^p_k$	–	**3.6434**	P
$u_k = -[0.2389\ 0.3973]x_k$	–	**3.5974**	LQR

Before presenting the results for $\zeta = 0.5$ and $\zeta = 0.2$ we comment the numbers in Table 5.2. First, we see that the initial convex optimization ranks the first-move P and PD controller over the PI and PID controller, i.e. $L^{\text{PD}}_{\text{wc}} = L^{\text{P}}_{\text{wc}} = 0.7171 < 0.7555 = L^{\text{PID}}_{\text{wc}} = L^{\text{PI}}_{\text{wc}}$. We also have that the closed-loop norms satisfy $||F_l(P,K)||^{PD}_2 = ||F_l(P,K)||^{P}_2 = 3.6627 < 3.9535 = ||F_l(P,K)||^{PID}_2 = ||F_l(P,K)||^{PI}_2$ for the first-move controllers. A closer look reveals that the open-loop optimization sets the derivative gain to zero, so that the PID controller equals the PI controller, and PD the P controller. This owes to the zero weight on the derivative output y^d.

The gain on the integrated output y^i is small (which is the same as a high time constant) in both the PID and PI case because of the low weight. If more integral effect is desired one could increase the weight on y^i, however, in this example we weight it low so that we can compare the different controllers from their respecive norms (since we have zero weight on y^d the norm of each controller is mainly decided by the weight on y^p, which is equal for all controllers).

After the closed-loop optimization the controllers are ranked as $J^*_{\text{LQR}} < J^*_{\text{PD}} < J^*_{\text{P}} <$

$J^*_{\text{PID}} < J^*_{\text{PI}}$. As expected, controllers with integral effect has a higher closed-loop norm than the P and PD controller. Further, we see that the PD controller comes very close to the LQR controller which measures both states. This is confirmed by a simulation of the system with a disturbance $x_0 = [1\ 0]^\top$ in the initial state. The different closed-loop optimal controllers perform according to their $\mathscr{H}_2$ norm as seen from Figure 5.1b.



(a) Closed-loop responses for $\zeta = 0.8$.

(b) Close-up of the responses for $\zeta = 0.8$.

Figure 5.1: Output resulting from a disturbance $x_0 = [1\ 0]^\top$ in the initial state ($\zeta = 0.8$). The low gain (high time constant) on the integrated output for the PID and PI controller results in an output that does not reach zero within the simulation time of 10 s.

A last comment on the data in Table 5.2 is that the first-move P controller equals $K_{LQR}C^\dagger = K_{LQR}\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = -0.3973$, as expected from Eq. (3.23) in Section 3.2. We proceed by presenting the optimization results for $\zeta = 0.5$ and $\zeta = 0.2$.

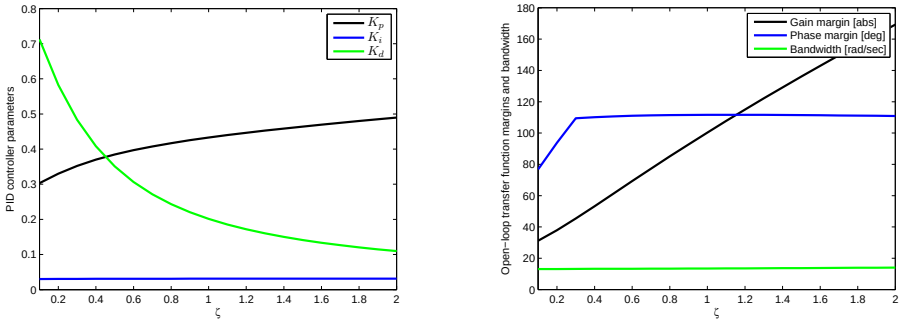
Table 5.3: Controllers for $\zeta = 0.5$

Control law	L_{wc}	$\|\|F_l(P,K^*)\|\|_2$	Controller
$u_k = -0.3956y^p_k - 0.0062y^i_k - 0y^d_k$	1.5234	4.0032	First-move PID
$u_k = -0.3846y^p_k - 0.0308y^i_k - 0.3509y^d_k$	–	**3.7353**	PID
$u_k = -0.3895y^p_k - 0y^d_k$	1.4774	3.8400	First-move PD
$u_k = -0.3575y^p_k - 0.3301y^d_k$	–	**3.6459**	PD
$u_k = -0.3956y^p_k - 0.0062y^i_k$	1.5234	4.0032	First-move PI
$u_k = -0.2619y^p_k - 0.0197y^i_k$	–	**3.9009**	PI
$u_k = -0.3895y^p_k$	1.4774	3.8400	First-move P
$u_k = -0.2414y^p_k$	–	**3.7966**	P
$u_k = -[0.3485\ 0.3895]x_k$	–	**3.6259**	LQR

Table 5.4: Controllers for $\zeta = 0.2$

Control law	L_{wc}	$\|\|F_l(P,K^*)\|\|_2$	Controller
$u_k = -0.3776y_k^p - 0.0063y_k^i - 0y_k^d$	6.3544	5.3924	First-move PID
$u_k = -0.3301y_k^p - 0.0303y_k^i - 0.5826y_k^d$	–	**4.1747**	PID
$u_k = -0.3728y_k^p - 0y_k^d$	6.2945	5.2894	First-move PD
$u_k = -0.3114y_k^p - 0.5617y_k^d$	–	**4.1157**	PD
$u_k = -0.3776y_k^p - 0.0063y_k^i$	6.3544	5.3924	First-move PI
$u_k = -0.1743y_k^p - 0.0087y_k^i$	–	**5.2684**	PI
$u_k = -0.3728y_k^p$	6.2945	5.2894	First-move P
$u_k = -0.1599y_k^p$	–	**5.1585**	P
$u_k = -[0.5845\ 0.3728]x_k$	–	**4.0630**	LQR

From Tables 5.3 and 5.4 we see that as ζ gets smaller the controllers with derivative action gains an advantage over the ones without. For both $\zeta = 0.5$ and $\zeta = 0.2$ the controllers are ranked $J_{\text{LQR}} < J_{\text{PD}} < J_{\text{PID}} < J_{\text{P}} < J_{\text{PI}}$ by their closed-loop norm. As ζ decreases from 0.5 to 0.2 the proportional and integral gain lowers, while the derivative gain increases to compensate for the fall in loop phase. This is illustrated for the PID controller in Figure 5.2 underneath.



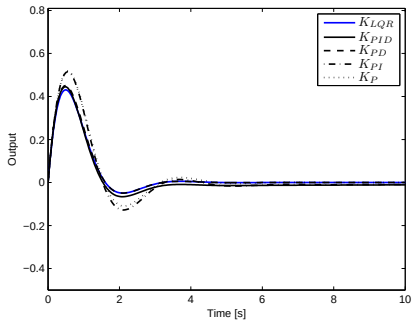
(a) $\mathscr{H}_2$-optimal PID controller gains.

(b) Bandwidth and margins.

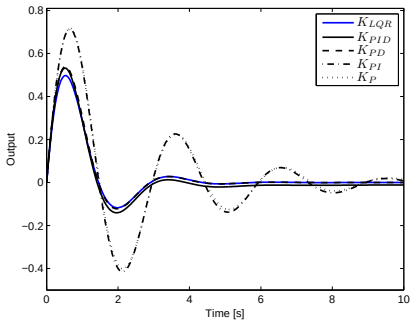
Figure 5.2: Plot of the $\mathscr{H}_2$-optimal PID controller gains as a function of ζ. The right plot shows the bandwidth, gain margin, and phase margin for the different PID controllers.

The closed-loop responses with the different controllers are presented in Figure 5.3. When $\zeta = 0.2$ the controllers without derivative action (P and PI) struggle with controlling the system, resulting in undesired oscillations in the output.

A motivation behind this example was to study how close the open-loop controller (K_0) is to the closed-loop $\mathscr{H}_2$-optimal controller (K^*) for the general second-order SISO system in (5.1). For the ζ values we have looked at the closed-loop optimiza-



(a) Closed-loop responses for $\zeta = 0.5$.



(b) Closed-loop responses for $\zeta = 0.2$.

Figure 5.3: Output resulting from a disturbance $x_0 = [1\ 0]^\top$ in the initial state ($\zeta = 0.5$ and $\zeta = 0.2$).

tion converges and we suspect that K_0 is fairly close to K^*. Table 5.5 confirms that the global minimum was found in all cases.[1] However, the PI controller runs into trouble as $\zeta \to 0$ and from Figure 5.3b we see that the closed-loop response is pretty bad for $\zeta = 0.2$. We pursue the question about the closeness of K_0 to K^* by considering the closed-loop norm for a range PI controller parameters.

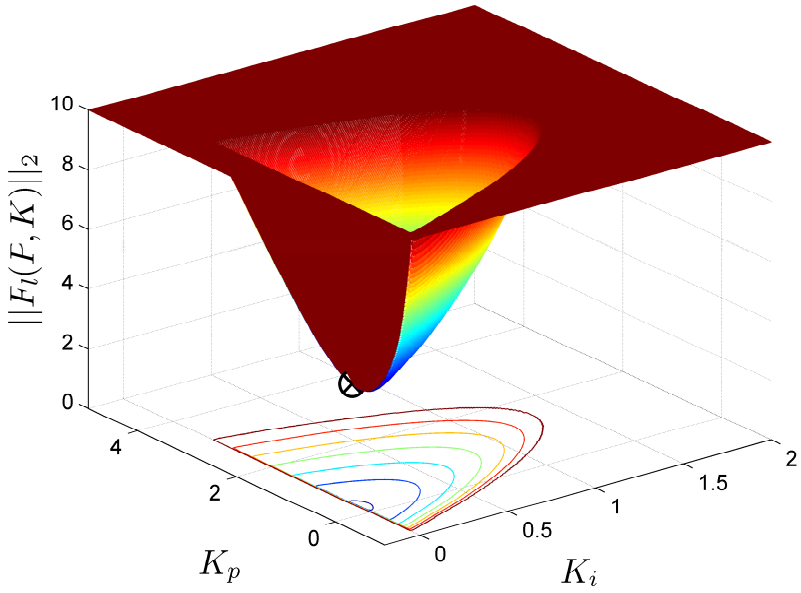
Figure 5.4 shows the closed - loop norm as a function of the PI controller gains, K_p and K_i, for $\zeta = 0.5$ and $\zeta = 0.2$. For values over 10 the closed-loop norm is replaced by 10. Most of the replaced values are infinity due to an unstable closed-loop system. The contours show the set of parameters that give a closed-loop norm below 10, i.e. give a stable closed-loop system. For $\zeta = 0.2$ this set shrinks considerably, along with the allowable values for K_0. In both cases the open-loop controller is close to the minimum and the closed-loop optimization converges to the global minimum, giving the $\mathscr{H}_2$-optimal gains in Tables 5.4 and 5.3 for the PI controller. With $\zeta \leq 0.05$ the open-loop controller supplied by Algorithm 3.1 fails to initialize the closed-loop optimization. However, these values give a system which is close to impossible to control with a PI controller, and P controller for that matter. And, the failing of Algorithm 3.1 is caused by a badly conditioned optimization problem.

	PID	PD	PI	P
$\zeta = 0.8$	√	√	√	√
$\zeta = 0.5$	√	√	√	√
$\zeta = 0.2$	√	√	√	√

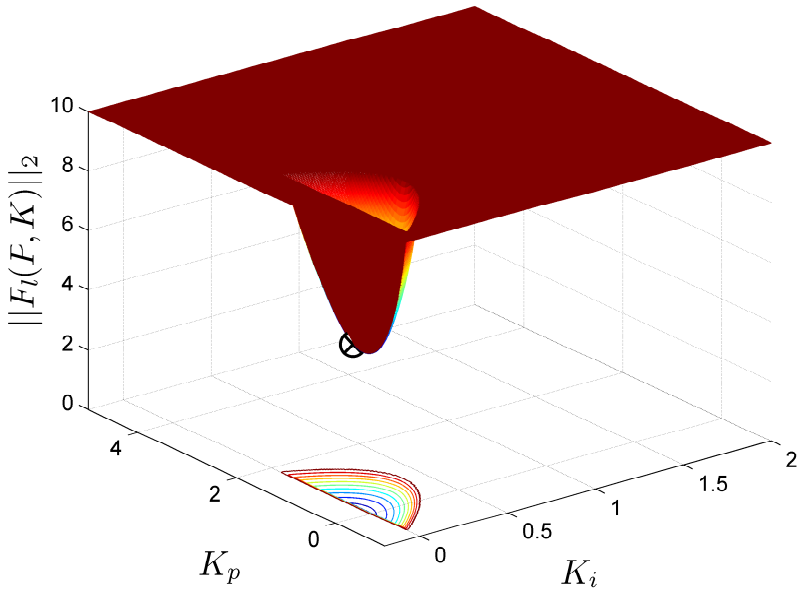
Table 5.5: Global optimality?

We conclude this example by noting that Algorithm 3.1 succeeds in finding a good starting point for the closed-loop $\mathscr{H}_2$ optimization when the system is of second-order SISO and controller is of type PID, PD, PI, or P.

[1] The global minima were found by calculating the $\mathscr{H}_2$ norm for all possible gains.



(a) $||F_l(P,K)||_2$ for $\zeta = 0.5$.



(b) $||F_l(P,K)||_2$ for $\zeta = 0.2$.

Figure 5.4: Plots of the $\mathscr{H}_2$ norm (closed-loop objective) for $\zeta = 0.5$ and $\zeta = 0.2$ and PI controller. The initial controller guess from the open-loop optimization (K_0) is marked by an encircled cross.

5.2 Study of the thermal/optical plant uDAQ28/LT

In this example we study the thermal-optical laboratory plant uDAQ28/LT, referred to as *udag* from here on [Jelenčiak et al., 2007].[2] The plant consists of 3 inputs and 2 ouputs, described in Tables 5.6 and 5.7.

Figure 5.5: The uDAQ28/LT

Table 5.6: Plant inputs

Input	Description	Range
u_1	Bulb voltage (heater & light source)	$0-5$V
u_2	Fan voltage (temperature decreaser)	$0-5$V
u_3	LED (diode) (light source)	$0-5$V

Table 5.7: Plant outputs

Output	Description	Unit
y_1	Temperature measurement	°C
y_2	Light intensity measurement	Not known[3]

[2]The thermal-optical laboratory plant (uDAQ28/LT) was developed and tested at the Faculty of Electrical Engineering and Information Technology in Bratislava by M.Huba and M. Kamensky in cooperation with the company Digicon (P. Kurcik). The plant is used for teaching purposes at several universities, including: FernUniversität in Hagen (prof. Gerke), University of Split (prof. Stipanicev), University of Ancona (prof. Longhi), and NTNU (prof. Skogestad). The plant will facilitate the course "TKP4140 - Process Control" at NTNU, next Fall.

[3]The signal from the photodiode used to measure light intensity is scaled in hardware.

As illustrated in Figure 5.6 on the right-hand side the *udaq* has three inputs available for control: a bulb, two LED's, and a fan. The bulb is a source for both light and heat. The LED's are primarily a light source as they give off little heat. The fan is used to lower the temperature by transporting heat out of the system. Using these inputs we seek a control scheme for setpoint control of our two measurements; temperature and light intensity.

The *udaq* also supports measurements of temperature outside the system (room temperature) and fan velocity. In addition, filtered measurements of the temperature and light intensity are available. These measurements can be used to improve control, e.g. via feed-forward of outside temperature. However, in this example we only take advantage of the unfiltered measurements in Table 5.7.

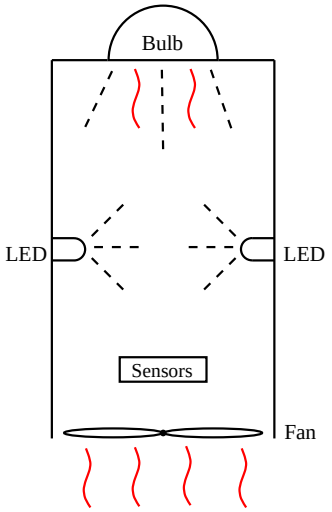


Figure 5.6: Schematic drawing of the uDAQ28/LT showing the three inputs and the placement of the sensors. The red, wavy lines illustrate heat and the dashed lines light emittance.

5.2.1 Model identification and scaling

A 5-state model for the thermal plant was identified in [Manum, 2008]. The identification was performed with a sampling time of $T_s = 0.5\,\text{s}$ using an inverse repeated input sequence around the operation point $u^* = [2\ 2\ 2]^\top$, with a room temperature of about 21 C°. The steady-state output, y^*, may change with changing disturbances such as a varying room temperature. Plant measurements indicate that $y^* \approx [38\ 22]^\top$ under the mentioned conditions.

A simulation with positive and negative unit steps in each input spread over a time span of 30 min show that the identified model is poorly scaled (the states was of magnitude 10^{-3} to 10^{-2}). Using the maximum variation in each state a diagonal scaling matrix is constructed. The proposed state scaling is $\bar{x} = Zx$, where $Z = \text{diag}(0.02, 0.01, 0.04, 0.02, 0.02)$ and x is the new state vector.[4] Note that all

[4] We here use the notation $A = \text{diag}(a)$, which is short for a diagonal matrix $A \in \mathbb{R}^{n\times n}$ with the elements of $a \in \mathbb{R}^n$ on its diagonal, i.e. $A = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & a_n \end{bmatrix}$.

properties of the system (e.g. eigenvalues) are preserved under any equivalence transformation $\bar{x} = Zx$, [Chen, 1999]. The inputs and outputs remain unscaled. Denoting the un-scaled model $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, we obtain the scaled model

$$x_{k+1} = \underbrace{Z^{-1}\bar{A}Z}_{A} x_k + \underbrace{Z^{-1}\bar{B}}_{B}\Delta u_k \tag{5.4}$$

$$\Delta y_k = \underbrace{\bar{C}Z}_{C} x_k + \underbrace{\bar{D}}_{D}\Delta u_k \tag{5.5}$$

with matrices

$$A = \begin{bmatrix} 0.9884 & -0.0053 & 0 & 0 & 0 \\ -0.0326 & 0.9710 & 0 & 0 & 0 \\ 0 & 0 & 0.4841 & 0.1691 & -0.0105 \\ 0 & 0 & -1.1736 & -0.2854 & 0.5170 \\ 0 & 0 & 0.4652 & 0.3695 & 0.8394 \end{bmatrix} \tag{5.6}$$

$$B = \begin{bmatrix} 0.0050 & -0.0050 & 0 \\ 0.0100 & -0.0200 & 0 \\ 0.1375 & 0 & 0.0425 \\ 1.2400 & 0 & 0.3750 \\ -0.4150 & 0 & -0.1250 \end{bmatrix} \tag{5.7}$$

$$C = \begin{bmatrix} 10.5011 & -0.0198 & 0 & 0 & 0 \\ 0 & 0 & 30.9550 & -3.0647 & 1.1134 \end{bmatrix} \tag{5.8}$$

$$D = 0_{n_y \times n_u} \tag{5.9}$$

The model describes the plant behaviour around the nominal points u^* and y^*. With deviation variables $\Delta y = y - y^*$ and $\Delta u = u - u^*$, where $u = [u_1\ u_2\ u_3]^\top$ and outputs $y = [y_1\ y_2]^\top$, we have that

$$x_{k+1} = Ax_k + B\Delta u_k \tag{5.10a}$$

$$\Delta y_k = Cx_k \tag{5.10b}$$

where we have omitted the D matrix since it is full of zeros.

To increase the plant interactions we can couple the LED to the fan, so that $u_3 = \alpha \cdot u_2$. This makes the plant more difficult to control as the fan now influences both temperature and light intensity. Introducing the relationship

$$\Delta u = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & \alpha \end{bmatrix}}_{\tilde{B}} \Delta\tilde{u}, \quad \Delta\tilde{u} = \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \end{bmatrix}, \tag{5.11}$$

we can rewrite our model, so that the new (coupled) model becomes

$$x_{k+1} = Ax_k + B\tilde{B}\Delta\tilde{u} \tag{5.12a}$$

$$\Delta y_k = Cx_k \tag{5.12b}$$

The models in 5.10 and 5.12 will later in this example be referred to as *model 1* and *model 2*, respectively. We will set $\alpha = 1$, note that this makes the plant somewhat harder to control, but since the LED has such weak effect on the light intensity the plant should still be relatively easy to control. The plant interactions are discussed in more detail in the next section. We summarize the two models in Table 5.8 below.

Table 5.8: *udaq* models.

	# inputs	# outputs	Equation
model 1	3	2	(5.10)
model 2	2	2	(5.12)

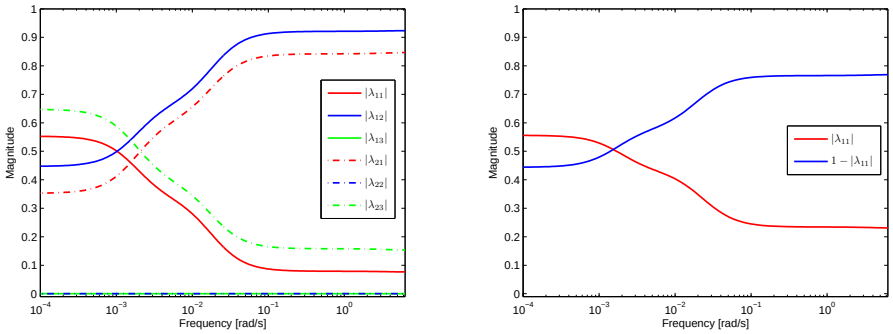
5.2.2 Analysis of dynamics

Let A_1 be the upper left 2×2 part of A, and let A_2 be the lower 3×3 part. We then have that the eigenvalues of A_1 are $\text{eig}(A_1) = \{0.996, 0.964\}$, and $\text{eig}(A_2) = \{0.020 \pm 0.041\,i, 0.999\}$. The eigenvalues are within the unit circle and the system is stable. Unfortunately the model is fairly stiff with eigenvalues close to zero and near the unit circle. This may pose some numerical problems in the ODE solver.

A closer look at the matrices shows that the temperature y_1 is determined by the states x_1 and x_2, while the light intensity y_2 is given by x_3, x_4, and x_5. From the eigenvalues we thus conclude that the light intensity dynamics are fast and temperature dynamics slow, as expected from a physical point of view. The eigenvalue belonging to x_5 shows that this state is very slow and further contributes little to the light intensity. The state has no physical meaning as the model is obtained by identification, but it is likely that it attempts to model a slowly varying nonlinearity.

The system is both controllable and observable. To study the *two-way* interactions in the plant we perform a relative gain array (RGA) analysis. Lets start by presenting the calculated RGA elements for *model 1* and *model 2* in Figure 5.7. The RGA elements are described in Table 5.9.

From the pairing rules in [Skogestad & Postlethwaite, 2005] we know that input-output pairs with an RGA element close to one is desired, and that pairing on negative or zero RGA elements should be avoided. Thus, for *model 1* we can exclude



(a) RGA elements for *model 1*.

(b) RGA elements for *model 2*.

Figure 5.7: Frequency-dependent RGA for *model 1* and *model 2*.

pairing on λ_{13} and λ_{22} (which is reasonable since pairing the fan with light intensity and LED with temperature obviously would not work). This leaves us with four RGA elements. A steady-state analysis ($\omega = 0$) indicate that pairing on λ_{11} and λ_{23} is the best alternative (i.e. bulb and LED to control temperature and light, respectively). However, with a bandwidth higher than 0.2 rad/s it is clearly best to pair on λ_{12} and λ_{21}, meaning that we (with decentralized control) would want to pair the bulb with light intensity and fan with temperature. These results are consistent with [Osuský & Hypiusová, 2009], where the topic is robust control design for the *udaq*.

The RGA elements for the square plant (*model 2*), in Figure 5.7b, suggest an off-diagonal pairing. So, we still want to pair the bulb with light intensity and the fan with temperature. Note that the RGA element $(1-\lambda_{11})$ is further from one, meaning that *model 2* has more two-way interaction than *model 1*.

Table 5.9: Description of RGA elements for *model 1* (left) and *model 2* (right).

	u_1	u_2	u_3
y_1	λ_{11}	λ_{12}	λ_{13}
y_2	λ_{21}	λ_{22}	λ_{23}

	u_1	u_2
y_1	λ_{11}	$1-\lambda_{11}$
y_2	$1-\lambda_{11}$	λ_{11}

5.2.3 Analysis of measurement noise

This section gives an analysis of the measurement noise present in the thermal-optical plant. A one hour long measurement sequence obtained around steady-state (when $u = u^* = [2\,2\,2]^\top$) is used as a basis for this analysis and it is assumed that the measurement noise is generated by an ergodic process.[5] A portion of the measurement sequence is presented below.

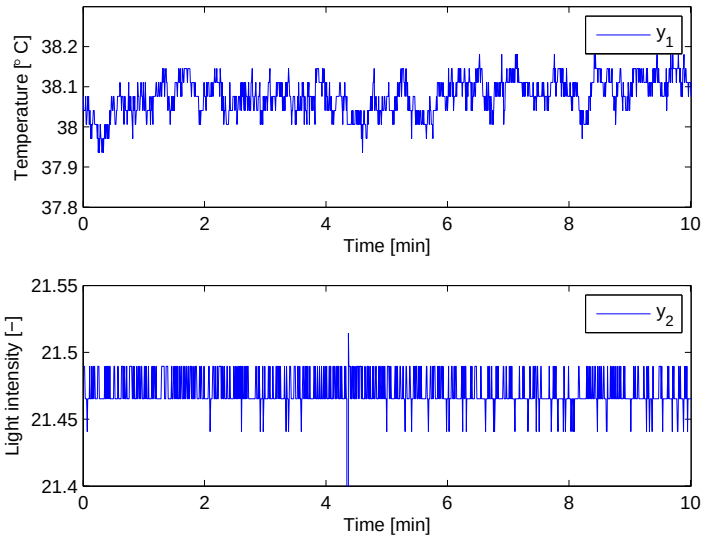
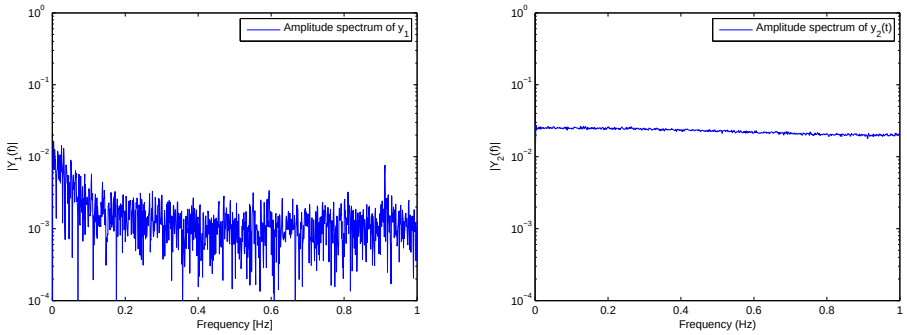


Figure 5.8: Measurement sequence of temperature (y_1) and light intensity (y_2) around steady-state.

Figure 5.8 clearly shows that the measurements are riddled with high-frequency noise. The peak in the light intensity are due to an disturbance in input voltage owing to internal variations in the power supply. A power spectral density (PSD) plot of the two signals shows the distribution of power at the different frequencies. The mean value of the sequences was removed before calculating the PSD using the fast Fourier transform (FFT). Note that the measurement sequence is sampled with a frequency of 2 Hz (sample time of 0.5 sec). From the Nyquist sampling theorem [Ljung, 1999] we know that frequencies below $1/(2T_s) = 1$ Hz (the Nyquist frequency) are captured, and that frequencies over 1 Hz are aliased and concieved as frequencies below 1 Hz.

[5] A stochastic process is said to be ergodic if its statistical properties (such as its mean and variance) can be deduced from a single, sufficiently long sample of the process.



(a) PSD of temperature measurement.

(b) PSD of light measurement.

Figure 5.9: PSD of measurement sequences.

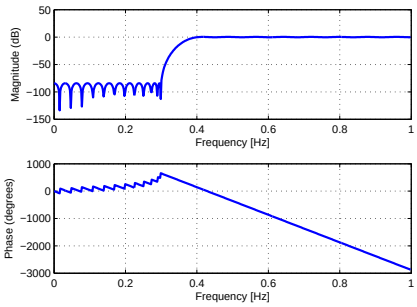
The PSD plot for the temperature measurement in Figure 5.9a tells us that most of the signal power lies at frequencies below 0.3 Hz. It is assumed that noise contributes to the signal at frequencies over 0.3 Hz and we note that a peak stands out at a frequency of about 0.91 Hz. With this assumption we can say that the temperature measurement noise is generated by a *zero-mean* stochastic process.

Figure 5.9b shows a rather flat power spectra for the light intensity measurement, which is expected due to the fast dynamics of the light. However, this makes it hard to determine if the power contribution comes from the plant response or measurement noise. Assuming that the noise is ergodic we can use the measurement sequence in Figure 5.9b to calculate a noise variance of $1.5 \cdot 10^{-4}$ for the light intensity.[6] As for the temperature measurement noise it is assumed that the light intensity noise has a mean of zero. Assuming otherwise would not help us since we by no means can detect a bias of the measurements using only the two measurement devices at hand.

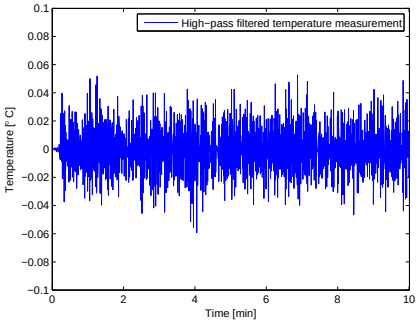
To analyze the effect of noise on the temperature measurement we try to high-pass filter the measurement sequence. This will remove the low frequencies and show the high-frequency part of the signal. An Equiripple high-pass FIR filter of order 56 was created using the *Matlab filter design & analysis tool* (fdatool). The filter parameters was chosen to attenuate frequencies below 0.3 Hz – 0.4 Hz, as shown by the frequency response in Fig. 5.10a.

A close-up of the high-pass filtered temperature measurement is given in Fig. 5.10b. From this (high-pass filtered) measurement we calculate a noise variance of approximately $1.6 \cdot 10^{-3}$ for the temperature measurement.

[6] The variance is calculated using the *var()* command in Matlab™. The standard deviation is found by taking the square root of the variance.



(a) Frequency response of the Equiripple high-pass filter.



(b) High-pass filtered temperature measurement.

From this short analysis we have gained knowledge of the measurement noise in the *udaq*. We know that the temperature measurement is more noisy than the light intensity, and we have observed input disturbances causing large peaks in the light intensity measurements. Table 5.10 shows the calculated variance and standard deviation of the noise on the two measurements.

Table 5.10: Noise properties for the *udaq* measurements.

	Temperature (y_1)	Light intensity (y_2)
Noise mean	0	0
Noise variance	$1.6 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$
Noise standard deviation	0.04	~ 0.012

5.2.4 Controller synthesis

We want to control the plant using MIMO PID control. We will consider the MIMO versions of the PID controller in 2.13 and the low-pass filtered PI controller in (2.15), i.e.

$$u = \left(K_p + K_i \frac{1}{s} + K_d \frac{s}{\epsilon s + 1} \right) y = K_p y^p + K_i y^i + K_d y^d, \qquad \text{(PID)} \quad (5.13)$$

$$u = \left(K_p \frac{1}{\epsilon s + 1} + K_i \frac{1}{s(\epsilon s + 1)} \right) y = K_p y^{p,f} + K_i y^{i,f}, \qquad \text{(low-pass filtered PI)} \quad (5.14)$$

The latter is included in this example to see if the low-pass filter improves control when we have measurement noise. As described in B.4 we augment *model 1* and *model 2* with the controller states and choose y^p, y^i, and y^d as outputs for the PID case, and $y^{p,f}$ and $y^{i,f}$ for the PI case.[7] The PID and PI controller are both of order $2 \cdot n_y = 4$ (since we low-pass filter the input to the PI controller). Using our knowledge from the noise analysis we set the low-pass filter coefficient $\epsilon = 2$, which attenuates noise at frequencies $f \geq 1/\epsilon = 0.5\,\text{Hz}$.

The PID and PI controller are tuned by using Algorithm 3.1 (with noise weights) where we first obtain the open-loop optimal controller and then improve it by performing a closed-loop optimization. The controllers are then compared to the discrete-time LQG controller. To be able to compare the controllers we will use the same weight matrices in the objective function and construct the LQG controller with integral action, as outlined in 2.3.8. The LQG controller is then of order $n_x + n_y = 5 + 2 = 7$. The objective function we attempt to minimize is

$$J = x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i \qquad (5.15)$$

with $Q = \text{diag}(25, 10^{-3}, 15, 5, 10^{-3})$, $R = R_1 = \text{diag}(0.1, 2, 1)$, $P = Q$, and a horizon length $N = 120$ (which equals 1 min). We have denoted the input weight matrix R_1 to emphasize that this weight belongs to *model 1*. For *model 2* we will use $R_2 = \text{diag}(1, 2)$. The state weight matrix Q is chosen to preferre states that affect the outputs most, e.g. x_1 and x_3. The bulb is the input with greatest effect on the states and we give it a low weight to assure that the controller has the freedom to use it. To keep the input effect low we weigh the fan and LED more, forcing them to be used only when needed. E.g. we do not want the fan to be used too much since it transports heat out of the system. The weights was chosen by analyzing the scaled state-space model and with experience from simulations using different weight matrices.

[7] We perform the augmentation of *model 1* and *model 2* with the controller states in continuous time. We then discretize the augmented models before using them to find optimal controller tunings. The models are brought to and from discrete-time using the Matlab™ functions *d2c()* and *c2d()*.

As mentioned we consider the state-space models in 5.10 and 5.12 augmented with the states of the PID and low-pass filtered PI controller. In the LQG case the model is augmented with the integrated outputs. This introduces new states which we have to weight by expanding Q. To fairly compare the controllers we weight the integrated outputs y^i (and $y^{i,f}$) equal in all cases. We include the new weights by expanding Q with $Q_i = \text{diag}(0.01, 0.01)$, so that

$$Q_{LQG} = Q_{PI} = \begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}. \tag{5.16}$$

The new states, representing the integrated outputs, are the same for the LQG and PID controller. However, for the low-pass filtered PI controller these states represent a low-pass filtered version of the integrated outputs and we cannot expect an exact comparison between the controllers. Further, the PID controller introduces states for the derivative outputs y^d, in addition to y^i. These states are given a weight of zero so that they do not contribute to the objective function. This means that we do not constrain the use of derivative action through Q. Defining $Q_d = \text{diag}(0, 0)$ we have that

$$Q_{PID} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & Q_d \end{bmatrix} = \begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.17}$$

We choose the disturbance and noise weight matrices to be $W = I$ and $V = I$. The noise weight $E\{w_n w_n^\top\} = V$ was found in the noise analysis, but to guard against model uncertainty, input disturbances, and unknown noise we set $V = I$.[8] We summarize the weight matrices for *model 1* and *model 2* in Tables 5.11 and 5.12.

Table 5.11: Weight matrices for *model 1*

	Q	$R = R_1$	P	W	V
LQG	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	I	I
PID	$\begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & Q_d \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & Q_d \end{bmatrix}$	I	I
PI	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	I	I

[8] The weights from the noise analysis, $V = \text{diag}(1.6 \cdot 10^{-3}, 1.5 \cdot 10^{-4})$, give almost identical controllers as for $V = I$. However, reducing V increases the controller gains and hence the noise amplification in the controllers, in agreement with our intuition and (3.31). This has been confirmed by optimizations and simulations with low values for V.

Table 5.12: Weight matrices for *model 2*

	Q	$R = R_2$	P	W	V
LQG	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	I	I
PID	$\begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & Q_d \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} Q & 0 & 0 \\ 0 & Q_i & 0 \\ 0 & 0 & Q_d \end{bmatrix}$	I	I
PI	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0 \\ 0 & 2 \end{bmatrix}$	$\begin{bmatrix} Q & 0 \\ 0 & Q_i \end{bmatrix}$	I	I

Following the procedure in Section 3.2 we specify the measurement candidates to be used in the open-loop optimization of the PID and PI controller. Assuming that all inputs are measured we choose the measurement candidates to be $y_c = [y_0, u]^\top$, where $u = [u_0, u_1, \ldots, u_{N-1}]^\top$ and N is the horizon. y_0 represent the outputs of the augmented system at time $k = 0$, i.e. y_0^p, y_0^i, and so on. With $d = x_0$ we get

$$G^y = \begin{bmatrix} D_{\text{PID}} & 0_{n_u \times n_u(N-1)} \\ I_{n_u} & 0 \\ 0 & I_{n_u \times (N-1)} \end{bmatrix} u, \quad G_d^y = \begin{bmatrix} C_{\text{PID}} \\ 0_{n_u \times (n_x + 2n_y)} \end{bmatrix} d, \tag{5.18}$$

where C_{PID} and D_{PID} belongs to the respective model augmented with the PID controller. These matrices are replaced with C_{PI} and D_{PI} in the PI controller case. As before we use 3.5 and 3.6 to find J_{uu} and J_{ud}. The noise and disturbance weight for the measurement candidates are:

$$W_d = W, \quad W_{n^y} = \begin{bmatrix} V \\ 0_{n_u \cdot N} \end{bmatrix}. \tag{5.19}$$

We have now performed steps 1-4 in Algorithm 3.1 and we can use the open-loop model $y_c = G^y u + G_d^y d$ to find the optimal measurement combination H. From the measurement combination H we obtain the "first-move" controller (first invariant from H) which we enhance by performing a closed-loop optimization

$$\min_K J_{cl} = ||F_l(P, K)||_2,$$

with P representing the generalized plant (Fig. 2.5) of the augmented model and K the constant gain matrix. The results from the last two steps of the algorithm is summarized in Table 5.13 below. The optimal objective function value of the open-loop optimization, $J_{ol}^* = ||H\tilde{F}||_F$, is presented for the PID and PI controller.

Table 5.13: Objective function values for *model 1* and *model 2*.

	Controller	$J^*_{ol} = \|\|H\tilde{F}\|\|_F$	$J^*_{cl} = \|\|F_l(P,K^*)\|\|_2$
model 1	LQG	–	73.91
	PID	139.79	79.70
	PI	607.96	96.17
model 2	LQG	–	63.28
	PID	140.07	68.35
	PI	611.16	83.90

The LQR law and the two closed-loop optimal controllers for *model 1* are:

$$\text{LQR:}\quad u_k = -\begin{bmatrix} 2.3780 & -0.2356 & 0.7140 & 0.2226 & 0.0882 \\ -8.0456 & 0.6889 & 0.1699 & 0.0824 & 0.1263 \\ -6.9993 & 0.7409 & 0.2479 & 0.1562 & 0.3237 \end{bmatrix} \hat{x}_k - \begin{bmatrix} 0.0116 & 0.0320 \\ -0.0436 & 0.0052 \\ -0.0311 & 0.0065 \end{bmatrix} y^i_k,$$

$$\text{PID:}\quad u_k = -\begin{bmatrix} 0.2332 & 0.0027 \\ -0.7930 & -0.0042 \\ -0.7065 & -0.0193 \end{bmatrix} y^p_k - \begin{bmatrix} 0.0278 & 0.0384 \\ -0.0821 & 0.0491 \\ -0.0692 & 0.1296 \end{bmatrix} y^i_k - \begin{bmatrix} 0.3553 & 0.0235 \\ -0.7089 & 0.0987 \\ -1.0523 & 0.2730 \end{bmatrix} y^d_k,$$

$$\text{PI:}\quad u_k = -\begin{bmatrix} 0.1345 & 0.0417 \\ -0.6432 & -0.0011 \\ -0.5109 & 0.0007 \end{bmatrix} y^{p,f}_k - \begin{bmatrix} 0.0187 & 0.0119 \\ -0.0618 & 0.0237 \\ -0.0525 & 0.0615 \end{bmatrix} y^{i,f}_k,$$

The LQR law and the two closed-loop optimal controllers for *model 2* are:

$$\text{LQR:}\quad u_k = -\begin{bmatrix} 2.8647 & -0.2288 & 0.7234 & 0.2372 & 0.1340 \\ -8.6805 & 0.7258 & 0.2189 & 0.1094 & 0.1750 \end{bmatrix} \hat{x}_k - \begin{bmatrix} 0.0168 & 0.0320 \\ -0.0486 & 0.0066 \end{bmatrix} y^i_k,$$

$$\text{PID:}\quad u_k = -\begin{bmatrix} 0.2790 & -0.0006 \\ -0.8667 & -0.0108 \end{bmatrix} y^p_k - \begin{bmatrix} 0.0355 & 0.0552 \\ -0.0957 & 0.0829 \end{bmatrix} y^i_k - \begin{bmatrix} 0.3167 & 0.0604 \\ -0.9657 & 0.1717 \end{bmatrix} y^d_k,$$

$$\text{PI:}\quad u_k = -\begin{bmatrix} 0.1873 & 0.0414 \\ -0.6776 & 0.0004 \end{bmatrix} y^{p,f}_k - \begin{bmatrix} 0.0235 & 0.0198 \\ -0.0683 & 0.0384 \end{bmatrix} y^{i,f}_k,$$

With $W = V = I$ the steady-state Kalman filter gain becomes

$$K_f = \begin{bmatrix} 0.0944 & -0.0000 \\ -0.0086 & -0.0000 \\ -0.0000 & 0.0314 \\ 0.0000 & 0.0005 \\ 0.0000 & 0.0253 \end{bmatrix} \tag{5.20}$$

for both *model 1* and *model 2* (independent of the B-matrix).

5.2.5 Implementation

The controller synthesis gave us a PID controller on the form:

$$u_k = K_p y_k^p + K_i y_k^i + K_d y_k^d \tag{5.21}$$

To implement this discrete controller we need to discretize the expressions $y^i = (1/s)y$ and $y^d = \frac{s}{\epsilon s+1} y$. Using the bilinear transform, i.e. the approximation $s = \frac{2}{T_s}\frac{z-1}{z+1}$, with the sampling time $T_s = 0.5\,\text{s}$ we get

$$y_k^i = G^i(z) \cdot y_k, \quad G^i(z) \triangleq \frac{T_s}{2}\frac{1+z^{-1}}{(1-z^{-1})}, \tag{5.22}$$

$$y_k^d = G^d(z) \cdot y_k, \quad G^d(z) \triangleq \frac{2}{T_s}\frac{1-z^{-1}}{(1+2\epsilon/T_s)+(1-2\epsilon/T_s)z^{-1}}. \tag{5.23}$$

The discrete PID controller can then be implemented as

$$u_k = \left(K_p + K_i G^i(z) + K_d G^d(z)\right) y_k. \tag{5.24}$$

The PI controller with a low-pass filter is discretized in the same manner. Defining the discrete low-pass filter as

$$G^f \triangleq \frac{1+z^{-1}}{(1+2\epsilon/T_s)+(1-2\epsilon/T_s)z^{-1}} \tag{5.25}$$

gives us the relations

$$y_k^{p,f} = G^f(z) \cdot y_k \tag{5.26}$$

$$y_k^{i,f} = G^i(z) \cdot G^f(z) \cdot y_k, \tag{5.27}$$

and the discrete PI controller is implemented as

$$u_k = \left(K_p + K_i G^i(z)\right) G^f(z) \cdot y_k. \tag{5.28}$$

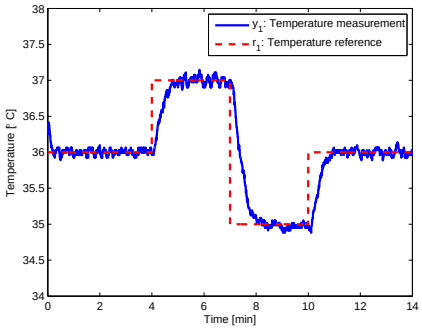
The discrete LQG controller is implemented as outlined in Chapter 2.3.8, by combining a discrete steady-state Kalman filter (2.3.4) and discrete LQR 2.3.2. The controllers (LQG, PID, and PI) are all implemented as 2-degree controllers with the reference signal entering only at the integrated output. As discussed in Chapter 2.2 this setup avoids derivative and proportinal kick when the reference signal is stepped. The diagrams in Appendix D.2 show how the discrete controllers are implemented in Simulink™.

5.2.6 Experiments

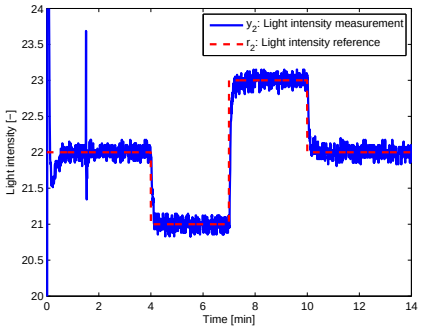
The *udaq* experiments are divided into two parts, one for *model 1* and one for *model 2*. These parts consist of three runs, one for each controller, resulting in a total of six data sets.

The experimental results for the case with three inputs (*model 1*) are presented first, before the case with two available inputs (*model 2*). To ease comparison of the controllers' performance we gather the plant responses in one figure. The controller outputs for both cases (*model 1* and *model 2*) are then presented together in one figure. This makes it more easy to study how much the increased interaction (in *model 2*) affects control. The results are discussed in the next section.

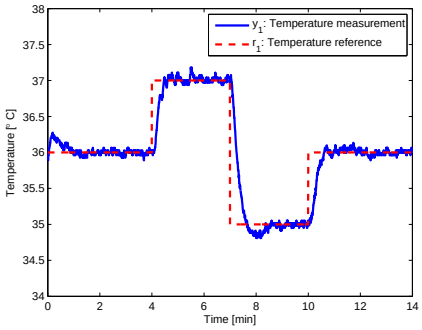
Before the experiments are run the plant is driven towards steady-state with 2V on all inputs. When the plant is sufficiently close to the operating point where the model was identified and controller designed for, control is turned on. This prevents input saturation during warmup. Since anti-windup is not implemented we only consider moderately large reference steps, keeping the inputs within the constraints and the states inside the operating region.



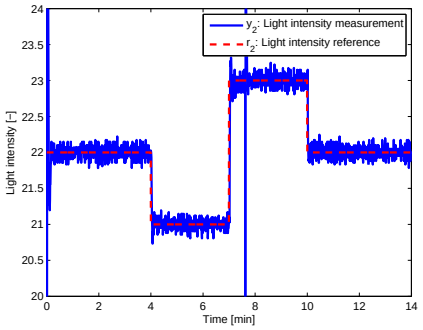
(a) Temperature measurement (LQG)



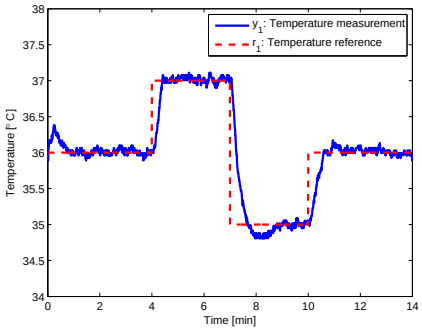
(b) Light intensity measurement (LQG)



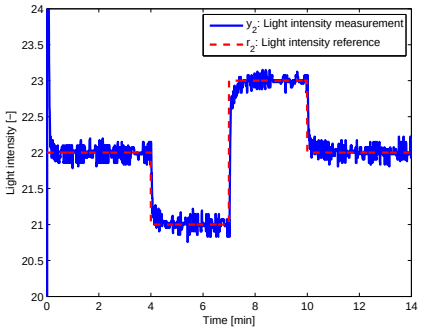
(c) Temperature measurement (PID)



(d) Light intensity measurement (PID)

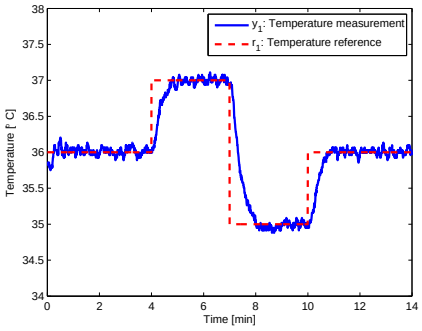


(e) Temperature measurement (PI)

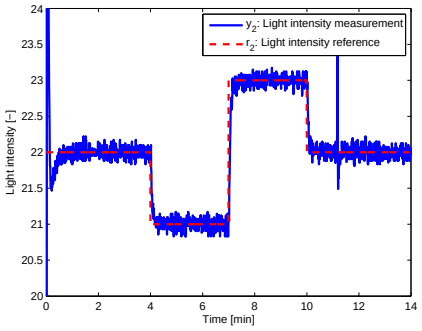


(f) Light intensity measurement (PI)

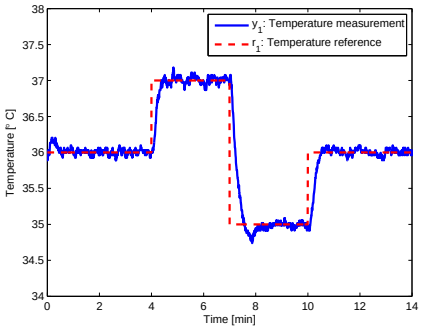
Figure 5.10: Temperature and light intensity measurements from the *udaq* using LQG, PID, and PI control (*model 1*, three inputs available). The reference signal is represented by a dashed, red line.



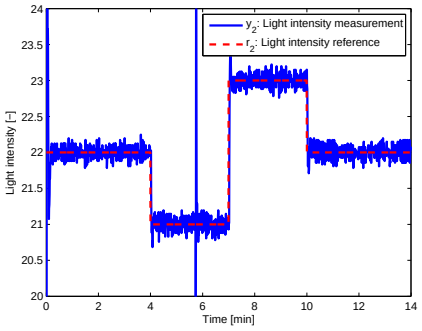
(a) Temperature measurement (LQG)



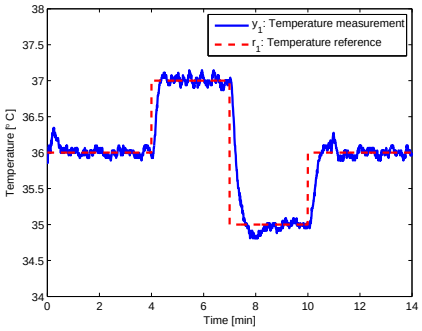
(b) Light intensity measurement (LQG)



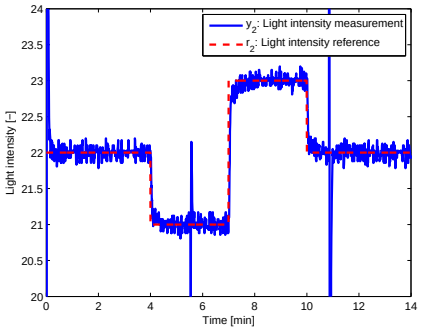
(c) Temperature measurement (PID)



(d) Light intensity measurement (PID)

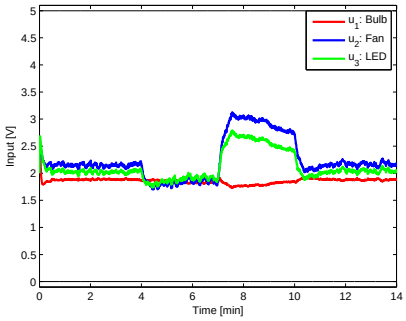


(e) Temperature measurement (PI)

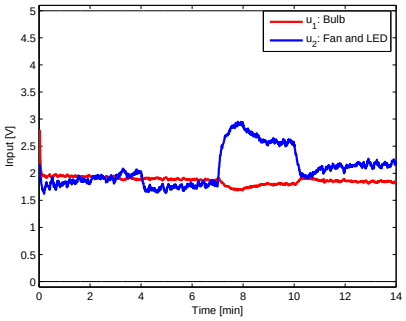


(f) Light intensity measurement (PI)

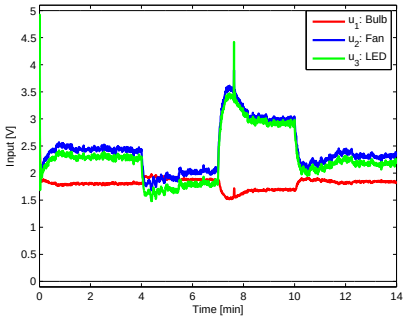
Figure 5.11: Temperature and light intensity measurements from the *udaq* using LQG, PID, and PI control (*model 2*, two inputs available). The reference signal is represented by a dashed, red line.



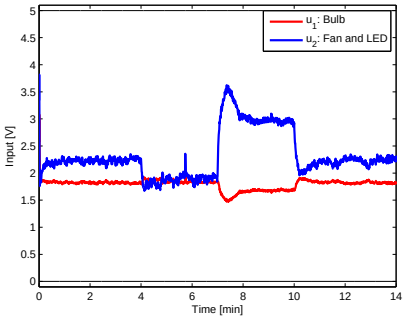
(a) LQG controller outputs (*model 1*)



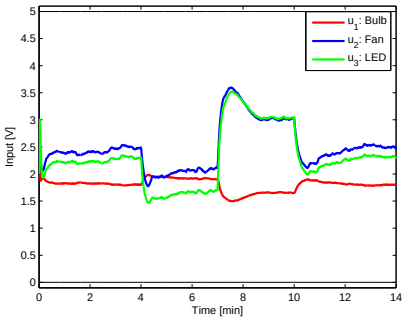
(b) LQG controller ouputs (*model 2*)



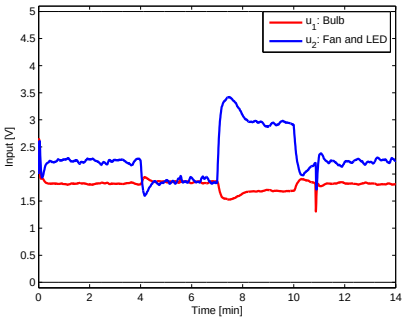
(c) PID controller outputs (*model 1*)



(d) PID controller outputs (*model 2*)



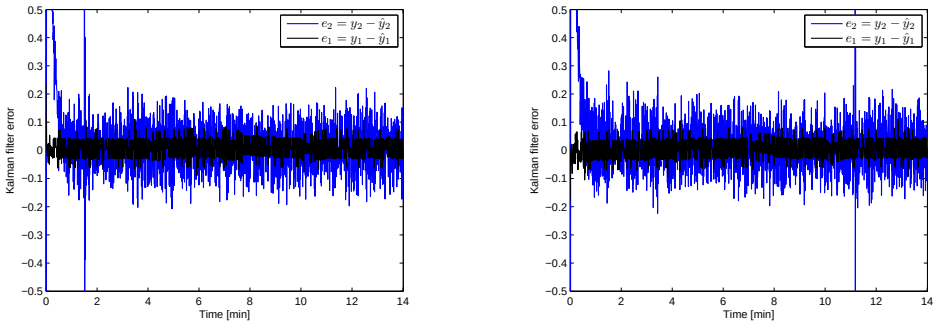
(e) PI controller outputs (*model 1*)



(f) PI controller outputs (*model 2*)

Figure 5.12: Controller outputs for the two cases (*model 1* and *model 2*) of LQG, PID, and PI control of the *udaq*.

The Kalman filter estimation error $e = y - \hat{y}$ for *model 1* and *model 1* is plotted in Figure 5.13 below.



(a) Estimation error with *model 1*

(b) Estimation error with *model 2*

Figure 5.13: Kalman filter estimation error with LQG control of the *udaq*.

To measure robustness and performance we have included a plot of the maximum singular values of the sensitivity functions for LQG, PID, and PI control of *model 2*. We have denoted the sensitivity functions $S_{LQG} = (I + GK_{LQG})^{-1}$, where G represent *model 2* and K_{LQG} the LQG controller. We use the same notation for S_{PID} and S_{PI}, and plot the maximum singular values $\bar{\sigma}(S)$ in Figure 5.14.

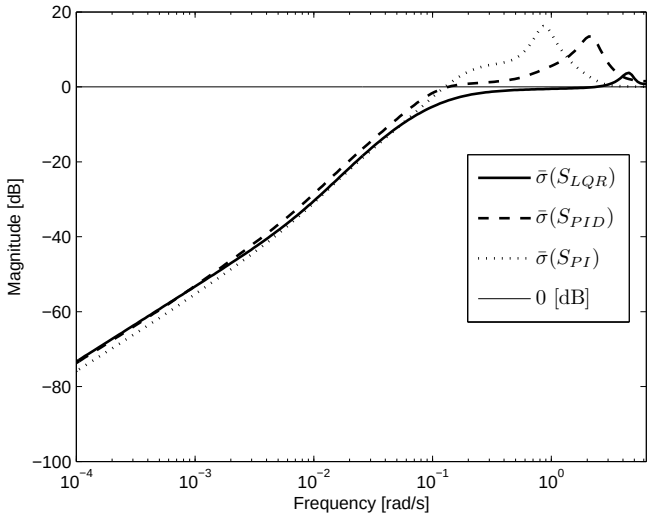


Figure 5.14: The maximum singular value of the sensitivity function with LQG, PID, and PI control of *model 2*.

5.2.7 Discussion of simulation results

Here we provide a discussion of the simulation results presented in the previous section. The controllers are measured up against eachother and we discuss in what degree the optimally tuned PID and PI controller performs in this practically example.

First of all we observe that the removal of one degree of freedom, i.e. the removal of u_3, giving *model 2*, has little affect on the controllers' performance. Consequently, we will consider the results for the case with three inputs (*model 1*), as they apply to the two-input case (*model 2*) aswell.

Second, the irregularity in input voltage gives spikes in the measurements. This disturbance shows up especially in the light intensity measurement because of the fast dynamics, see for example Figures 5.10b and 5.10d. The temperature dynamics is slow and filters out the spikes, hence we do not observe them in the temperature measurement plots. The input plots show the controller outputs and not the real voltage applied to the *udaq*, therefore the spikes does not show in these plots.

We start our comparison by studying the performance of the LQG controller, using Figures 5.10a, 5.10b, and 5.12a. The LQG controlled system follows both reference signals without overshooting, and the light intensity response is faster than the temperature, as expected. The spike in input voltage, presenting itself just before reaching two minutes in Fig. 5.10b, is filtered out and does not affect control worth mentioning. The controller outputs is reasonable smooth, as desired. We also see from Fig. 5.13 that the Kalman filter estimation error in both cases converges fast and then follows the measurement noise.

The PID controller, presented by Figures 5.10c, 5.10d, and 5.12c, does not filter the measurements (only the derivative part is filtered) and the controller outputs are noisier than with LQG control. The PID controller performs more aggressively than the LQG controller, reaching the reference faster at the cost of a small overshoot. The same holds for the PI controller, which overshoots even more when the temperature reference is stepped (Figures 5.10a, 5.10c, and 5.10e). Comparing with Figures 5.10e, 5.10f, and 5.12e we see that the PI controller also gives fast temperature control compared to the LQG. We further observe from the light intensity measurement (Figure 5.10f) that the PI controller applies smoother controller outputs (Figure 5.12e) than both the LQG and PID controller. The result is less noise amplification and a better light intensity response. Except from this the controller outputs for the PID and PI controller are similar in amplitude.

The maximum singular values in Figure 5.14 tells us that the closed-loop bandwidth (where $\bar{\sigma}(S)$ crosses $1/\sqrt{2} \approx -3\,\text{dB}$ from below) is a little bit higher for the LQG controller, but closely followed by the PID and PI controller. $\bar{\sigma}(S_{PID})$ and $\bar{\sigma}(S_{PI})$ both have a peak of over 12 dB, exceeding the recommended bound on $\bar{\sigma}(S) \leq 2 \approx 6\,\text{dB}$ [Skogestad & Postlethwaite, 2005]. In a real application it would be desirable to lower these peaks, gaining more robustness against model uncertainty.

Better performance can be obtained by further tuning of the controllers, changing Q, R, W, and V. However, this example was constructed not to obtain "perfect" control, but to compare the controllers tuned by Algorithm 3.1 with the LQG controller. This example proves that the tuning algorithm can be used in practice to obtain good controller parameters. It is by far more easy to tune a MIMO controller, e.g. a MIMO PID controller, by specifying the weight matrices, to the contrary of tuning each PID controller individually. Take for example a 4-input 4-output system to be controlled with a MIMO PID controller. Lets say that each controller has 3 parameters, resulting in a total of $3 \cdot 16 = 48$ parameters to find. Throw some interaction into the mix and this tuning problem becomes difficult, to say the least. The tuning algorithm on the other hand requires 4 output weights, 4 input weights, and the specification of disturbance and noise weights, W and V.

The controller structure is an important part of any controller design. As this example shows, a low-pass filtered PI controller may perform better than a PID controller with a low-pass filter only on the derivative part. The lesson to be learned is that controller tuning is not the only important part of control design, choosing the right controller structure is just as important.

5.3 Robustness properties of LQG

As stated and proved by example in [Doyle, 1978] the LQG controlled system with a combined Kalman filter and LQR control law has no guaranteed stability margins.[9]

As pointed out in [Athans et al., 1981] and by many other authors the LQR controlled system (with all states available and no stochastic inputs) has good stability margins when R is diagonal, with a gain margin equal to infinity, a gain reduction margin (lower gain margin) of 0.5, and a phase margin of 60°. In fact, when V is diagonal the Kalman filter enjoys the same margins as the LQR. However, when coupled together there are no guarantees for good robustness properties.

It seems that this knowledge often is forgotten in the control community. Model errors are inevitable and controller design using the separation theorem may give an unstable system in practice. The confusion is strenghtened by simulations using a perfect model, which give no indication of instability. The following example serves as a reminder to LQG designers that the robustness properties of a LQG controlled system always should be checked.

5.3.1 Example: LQG control of an unstable SISO system with one RHP-zero

In this example we try to design an LQG controller for the following plant

$$g_1(s) = \frac{-1}{s-1}, \quad g_2(s) = \frac{s-0.5}{s+0.5}, \quad G(s) = g_1(s) \cdot g_2(s), \tag{5.29}$$

where $G(s)$ is the transfer function of the plant. The plant has poles at $s = \{-0.5,\ 1\}$ and a RHP-zero at $s = 0.5$. The proximity of the RHP-zero to the RHP-pole makes the plant difficult to control. The control goal in this example is to keep the output at zero by counteracting measurement noise and input disturbances. A minimal realization (i.e. controllable and observable model) of the plant is

$$A = \begin{bmatrix} 0.5 & 0.5 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 0.5 \end{bmatrix}, \quad D = 0. \tag{5.30}$$

[9] *Stability margins* or simply *margins* refers to the (upper and lower) *gain margin* and (upper and lower) *phase margin* of a system. The stability margins tells us how close a stable closed-loop system is to instability. More precisely, the gain and phase margins of the open-loop system tells us by how much the gain and phase of the open-loop system can be increased or decreased before the closed-loop system becomes unstable. The phase margin translates directly to the *delay margin*, which is the maximum time delay that can be introduced to the loop before the closed-loop system becomes unstable.

Including input disturbance and measurement noise gives the model

$$\dot{x} = Ax + Bu + Bw_d \tag{5.31}$$
$$y = Cx + w_n \tag{5.32}$$

Chosing the weights $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$ and $R = 1$ gives the LQR law $K_r = [4.70\ 3.70]$. With both states available the closed-loop poles become $s = \{-3.20, -1.00\}$ using the LQR law. The Kalman filter is constructed with the power spectral density weights $W = 1$ and $V = 0.01$, with W and V as defined in Section 2.3.6. With these weights the Kalman gain is $K_f = [-22.10\ -22.10]$, giving the Kalman filter poles at $s = \{-10.05, -0.50\}$. By the separation theorem we combine the LQR law and Kalman filter as shown in the Simulink™ schemes in Appendix D.1.

The LQG transfer function $K_{LQG}(s)$ is given by Eq. (2.40) and the open-loop transfer function is $G_{ol}(s) = -K_{LQG}(s) \cdot G(s)$ (using negative feedback). G_{ol} has poles at $s = \{-21.1601, 5.9087, 1.0000, -0.5000\}$ and we imidiately identify two RHP-poles. We investigate the closed-loop stability by using the *Nyquist stability criteria* for negative feedback systems:

> For the closed-loop system to be stable, the open-loop system $G_{ol}(s)$ with N_p poles in the right-half plane must have N_p counterclockwise encirclements around the point $(-1+0j)$ when s runs clockwise through a contour which encompasses the right-half of the complex plane. [Balchen et al., 2003]

With two RHP-poles we thus need two counterclockwise rotations around the point $(-1+0j)$. The Nyquist diagram of the open-loop transfer function $G_{ol}(s)$ is plotted in Figure 5.15.

We observe that we have two counterclockwise rotations and conclude that the closed-loop system is stable according to the Nyquist stability theorem. The Nyquist diagram also provide information about the system margins. From the diagram in Fig. 5.15 we read a phase margin of $p_m = 24.7°$ at the frequency $\omega_p = 5.92\,\text{rad/s}$. This can be translated into a (time) delay margin of $\theta_{max} = \frac{\pi p_m}{180\omega_p} \approx 0.07\,\text{s}$ (assuming that p_m is given in degrees and ω_p in rad/s). In other words, adding a time delay higher than 0.07 s will render the closed-loop system unstable. Unmodeled dynamics are often introduced via time delays, and with such low margins the controller would probably not work in practice.

Tuning the LQR and Kalman filter more agressively (by increasing $\frac{||Q||}{||R||}$ and $\frac{||W||}{||V||}$, respectively) would result in even lower margins. For example; with $V = 0.001$ and

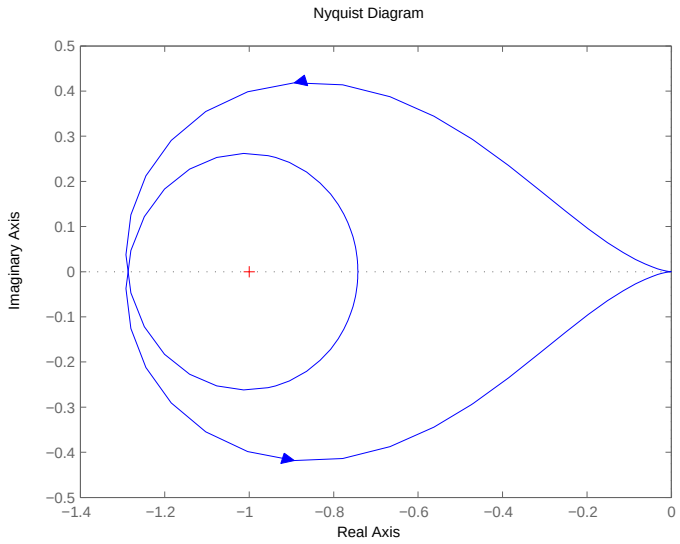
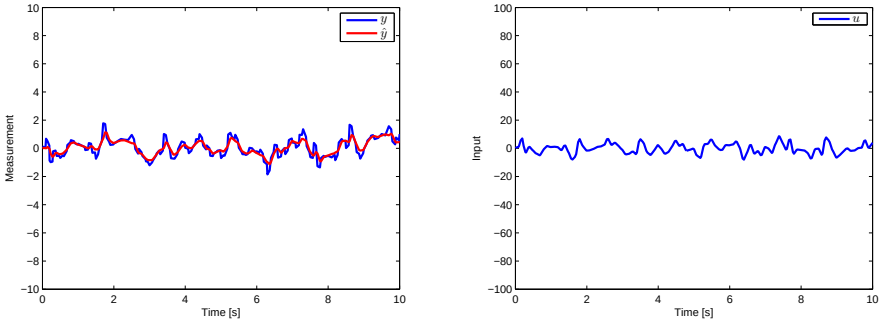


Figure 5.15: Nyquist diagram of $G_{ol}(s)$

$R = 0.1$, Q and W as before, the phase margin becomes 25.9° at the frequency 15.3 rad/s, which translates to a delay margin of poor 0.0295 s.

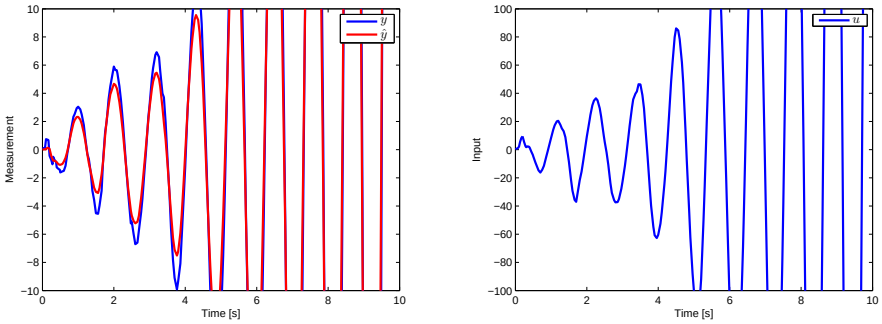
The plant with the designed LQG controller is simulated for 10 s with a measurement noise power $E\{w_n^2\} = 0.01$ and input noise power $E\{w_d^2\} = 0.01$. Figure 5.16 shows the response without time delay, i.e. perfect model. Next, the system is simulated with a time delay $\theta_d = 0.08$ s, which exceeds the calculated delay margin of 0.07 s. The response with time delay is presented in Figure 5.17.

From Fig. 5.17 it is evident that even small model errors, contributing with extra phase lag, can make the system unstable. The moral of this example is that LQG designers must check margins for each specific case they work with. MPC designers, being less lucky in that they do not have access to the transfer function matrix of the controller, should at least check margins in the most important regions.



(a) Measurement and output estimate (b) Input usage

Figure 5.16: Simulation of LQG controlled plant without time delay. The plant is excited by measurement and input noise with power $E\{w_n^2\} = 0.01$ and $E\{w_d^2\} = 0.01$, respectively.



(a) Measurement and output estimate (b) Input usage

Figure 5.17: Simulation of LQG controlled plant with a time delay of 0.08 s. The plant is excited by measurement and input noise with power $E\{w_n^2\} = 0.01$ and $E\{w_d^2\} = 0.01$, respectively.

Chapter 6

Final discussion

After all the examples in Chapters 4 and 5 it now seems fit to provide the reader with a final discussion. The discussion deals with the results that has been obtained and the knowledge that has been gained through these examples.

With the theory from Chapter 2 in place we were able to review the recently proposed initialization scheme for the $\mathscr{H}_2$-optimal static output feedback problem [Manum et al., 2009]. The initialization scheme consists of formulating a convex problem using theory from self-optimizing control. By solving the convex program we find a static controller that in the open-loop sense is closest to the LQ-optimal state feedback controller. This controller is used as the starting point for a closed-loop $\mathscr{H}_2$ optimization with the goal to improve control. The $\mathscr{H}_2$ problem is formulated to represent the LQG problem, or LQR problem in the deterministic case, and we can specify performance weights in the same manner as with LQG or model predictive control.

After the review we put the $\mathscr{H}_2$-optimal static output feedback problem under the scope. Facing the new optimization problem we tried to answer the question of convexity. More precisely, we wanted to know *when* it is convex. The investigation started by recalling the advice from [Syrmos et al., 1997]; look at specific systems and exploit their structure when confronting a static output feedback problem. With this in mind we set out by looking at some interesting examples where the optimization problem becomes non-convex. The system in one of these examples was found by a *brute-force* algorithm which was designed to find SISO systems giving a non-convex problem. From these simple, but yet attractive examples we understood that the convexity property is decided by the system's closed-loop behaviour, which is an intricate function of the controller parameters, in addition to

the usual demand on controllability and observability. With the knowledge of this complexity and that some systems may give a non-convex optimization problem, we ended the search for necessary and sufficient conditions for convexity, and focused on the next important question; how good is the initial controller guess from the convex optimization?

With several local minima it is imperative to guess on a controller gain that is close to the global minimum, or else the optimization may converge to a suboptimal solution. In addition, the guess must be on a stabilizing controller. From Chapter 2.1 we know that in the discrete case there exists a stabilizing gain that minimizes the infinite-horizon LQ problem locally under certain conditions. We also know from [Syrmos et al., 1997] that a direct procedure for finding such a controller was unknown at that time.

This motivated Example 5.1, "$\mathscr{H}_2$-optimal control of a second-order system", where we investigated the initialization procedure for the general class of stable second-order SISO systems. We considered PID, PD, PI, and P control, and the example showed that for this class of systems the initial controller guess was close enough for the closed-loop optimization to converge to the global optimum in all cases. By plotting the $\mathscr{H}_2$ norm versus the controller parameters we confirmed that the initial controller guess was close to the globally optimal controller for the case with PI control. This gave us the confidence to move on to testing the algorithm on a more complicated system.

In Example 5.2 we applied the algorithm to the thermal/optical plant, uDAQ28/LT. This is a MIMO plant with three inputs and two noisy measurements. We obtained a 3-input 2-output model and a 2-input 2-output model with increased interactions for the plant. The interactions was analyzed using RGA and measurement noise studied using a long measurement sequence taken when the plant was close to the steady state. A MIMO PID and low-pass filtered MIMO PI controller was created and tuned using Algorithm 3.1. We also designed a discrete LQG controller (Kalman filter and LQR) for comparison. All controllers were given the same design weights; Q, R, W, and V.

With the controllers in place we conducted three experiments for each model, one per controller. The result from the experiments was that the MIMO PID and PI controller performed almost as good as the LQG controller, even being of three orders less than the LQG. Of the two controllers found using the algorithm the PI controller performed best because of the low-pass filtering of the measurements.

We ended the example by underlining that it always is important to use a controller structure that fits the problem at hand. We also noted that it is much simpler to tune a MIMO PID controller by specifying design weights over the more

conventional tuning method where each PID controller is tuned one at the time (and then retuned because of plant interactions).

The last example in Chapter 5, Example 5.3, stands out as it is not connected to the $\mathscr{H}_2$-optimal static output feedback problem in any way. The example is named "Robustness properties of LQG" and it was included in this thesis to remind the reader that even full order controllers such as the LQG may give unacceptable stability margins for some systems.

In the next chapter we summarize the results that have been obtained in this thesis. The thesis is then wrapped up by a list of suggestions for further research.

Chapter 7

Conclusion

The initialization scheme for the $\mathscr{H}_2$-optimal static output feedback problem [Manum et al., 2009] has been reviewed and tested in practice on several SISO and MIMO systems. We were motivated by the search for simple, low-order controllers, as an alternative to full-order model-based controllers. In this thesis we used the LQG controller as reference for comparing with the syntesized low-order controllers. Based on the examples we have given, we draw the conclusion that *the initialization scheme is suitable for many interesting systems.* This is the main conclusion of this thesis.

The value of the results in [Manum et al., 2009] is directly related to the convexity of the $\mathscr{H}_2$-optimal static output feedback problem. The simple counterexamples provided in this thesis show that the problem is non-convex in certain cases. From these examples we also observed that the convexity property, when the controller gains are the degrees of freedom in the optimization problem, is a function of the system's closed-loop dynamics. E.g. the set of stabilizing controller gains must be a connected set, if not, the optimization problem has more than one minima. However, from the investigation of the convexity we learned that the problem is convex for most SISO systems of order three or less.

On the examples considered the initialization procedure gave initial controller guesses sufficiently close to the global optimum for the closed-loop optimization reach it. With further research this procedure, which has the advantage that it is convex, may fill an empty space in available algorithms for static output feedback design.

MIMO controllers were syntesized and tested on the thermal/optical plant with

good results. The synthesized controllers had a lower order than the LQG controller, but still performed almost as good. We concluded this example by remarking that; the algorithm in [Manum et al., 2009] handles noise, gives acceptable control compared to the LQG controller, and can be utilized to tune MIMO controllers, such as the MIMO PID controller.

In the search for simple, low-order controllers several interesting observations were made. The author hopes these observations can be used as a starting point for further research on the initialization procedure.

The author is confident in that the static output feedback problem some time in the future again will entize the control community, if it is in the search for simple controllers or related to other problems.

7.1 Summary

The reader will find a list of the most important contributions from thesis below. The author has tried to sort them from most to least significant.

- The initialization procedure in [Manum et al., 2009] has been verified and tested on several examples.
- The convexity of the $\mathscr{H}_2$-optimal static output feedback problem has been analyzed, mainly through examples.
- The thermal/optical plant uDAQ28/LT was studied. MIMO controllers were synthesized using the algoritm in [Manum et al., 2009] and compared against the LQG controller by performing experiments.
- A Matlab™ framework implementing the theory from self-optimizing control and for solving Algorithm 3.1 was developed. The code may be useful in fascilitating later research.
- A literature study was conducted on the topics: static output feedback, PID, LQG and $\mathscr{H}_2$-optimal control, and self-optimizing control.

7.2 Future research

Here, the author shares his thoughts on what future research on the initialization scheme in [Manum et al., 2009] and the $\mathscr{H}_2$-optimal static output feedback problem may bring. The list presented below is intended for researches who may be interested in taking the ideas in [Manum et al., 2009] and this thesis further.

- By further classification of system-controller pairs that give a convex optimization problem, Algorithm 3.1 can be developed as a tuning tool for SISO and MIMO controllers. More importantly, for these system-controller pairs the algorithm is a complete static output design tool as it is.
- The algorithm can be extended to include constraints. One must then use the finite-horizon impulse-response formulation/approximation, and assure that the constraints are convex in the controller gains. Only then, can the algorithm compete with more advanced control techniques, such as MPC.
- Further investigate how the algorithm can be altered to make more robust controllers, e.g. by extending the performance criterion (objective function).

References

[Alstad & Skogestad, 2007] Alstad, V. & Skogestad, S. (2007). Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables. *Ind. Eng. Chem.*, 46, 846–853.

[Alstad et al., 2008] Alstad, V., Skogestad, S., & Hori, E. S. (2008). Optimal measurement combinations as controlled variables. *Journal of Process Control*, 19, 138–148.

[Athans et al., 1981] Athans, M., Sandell Jr, N., & Lehtomaki, N. (1981). Robustness results in linear-quadratic Gaussian based multivariable control designs. *IEEE Transactions on Automatic Control*, 26(1), 75–93.

[Balchen et al., 2003] Balchen, J. G., Andresen, T., & Foss, B. A. (2003). *Reguleringsteknikk*. Deparment of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU).

[Blondel et al., 1995] Blondel, V., Gevers, M., & Lindquist, A. (1995). Survey on the state of systems and control.

[Boyd & Vandenberghe, 2004] Boyd, S. & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[Brown & Hwang, 1997] Brown, R. G. & Hwang, P. Y. C. (1997). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Ltd.

[Chen, 1999] Chen, C.-T. (1999). *Linear System Theory and Design*. Oxford University Press.

[Cormen et al., 2001] Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction to algorithms*. MIT press.

[Doyle, 1978] Doyle, J. C. (1978). Guaranteed Margins for LQG Regulators. *IEEE Trans. Automatic Control*, 23, 756–757.

[Fu, 2004] Fu, M. (2004). Pole placement via static output feedback is NP-hard. *IEEE Transactions on Automatic Control*, 49(5), 855–857.

[Gilbert & Tan, 1991] Gilbert, E. & Tan, K. (1991). Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9), 1008–1020.

[Grant & Boyd, 2009] Grant, M. & Boyd, S. (2009). "CVX: Matlab Software for Disciplined Convex Programming (web page and software)". Available at: http://www.stanford.edu/ boyd/cvx/. (April 2009).

[He et al., 2000] He, J., Wang, Q., & Lee, T. (2000). PI/PID controller tuning via LQR approach. *Chemical Engineering Science*, 55(13), 2429–2439.

[Iwasaki & Skelton, 1994] Iwasaki, T. & Skelton, R. (1994). All controllers for the general $\mathscr{H}_\infty$ control problem: LMI existence conditions and state space formulas. *Automatica*, 30(8), 1307–1317.

[Iwasaki et al., 1994] Iwasaki, T., Skelton, R., & Geromel, J. (1994). Linear quadratic suboptimal control with static output feedback. *Systems & Control Letters*, 23(6), 421–430.

[Jelenčiak et al., 2007] Jelenčiak, F., Kurčík, P., & Huba, M. (2007). Thermal plant for education and training. *In Proceedings for ERK*, B, 318–321.

[Kalman, 1960] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1), 35–45.

[Keerthi & Gilbert, 1988] Keerthi, S. & Gilbert, E. (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2), 265–293.

[Levine & Athans, 1970] Levine, W. S. & Athans, M. (1970). On the determination of the optimal constant output feedback gains for linear multivariable systems. *IEEE Trans. Automatic Control*, 15, 44–48.

[Ljung, 1999] Ljung, L. (1999). *System Identification, Theory for the User*. Prentice Hall PTR.

[Manum, 2008] Manum, H. (2008). Thermal/Optical Plant. Identification and noise study of the uDAQ28/LT. Internal report, Department of Chemical Engineering, NTNU.

[Manum et al., 2007] Manum, H., Narasimhan, S., & Skogestad, S. (2007). A new approach to explicit MPC using self-optimizing control. Internal report, Department of Chemical Engineering, NTNU. Available at: http://www.nt.ntnu.no/users/skoge/publications/2007/.

[Manum et al., 2009] Manum, H., Skogestad, S., & Jäschke, J. (2009). Convex initialization of the $\mathscr{H}_2$-optimal static output feedback problem. *In Proceedings of the American Control Conference*, (pp. 1724–1729).

[Moerder & Calise, 1985] Moerder, D. & Calise, A. (1985). Convergence of a numerical algorithm for calculating optimal output feedback gains. *IEEE Transactions on Automatic Control*, 30(9), 900–903.

[Naidu, 2003] Naidu, D. (2003). *Optimal control systems*. CRC Press.

[Nocedal & Wright, 2006] Nocedal, J. & Wright, S. J. (2006). *Numerical Optimization*. Springer.

[Osuský & Hypiusová, 2009] Osuský, J. & Hypiusová, M. (2009). Robust Control Design for Thermo-Optical Plant uDAQ28/LT. In M. Fikar & M. Kvasnica (Eds.), *Proceedings of the 17th International Conference on Process Control '09* (pp. 341–345). Štrbské Pleso, Slovakia: Slovak University of Technology in Bratislava.

[Skogestad, 2003] Skogestad, S. (2003). Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control*, 13, 291–309.

[Skogestad, 2009] Skogestad, S. (2009). Feedback: Still the simplest and best solution. Available at: http://www.nt.ntnu.no/users/skoge/publications/2009/. IEEE Conference on Industrial Electronics and Applications.

[Skogestad et al., 2003] Skogestad, S., Halvorsen, I. J., Morud, J. C., & Alstad, V. (2003). Optimal selection of controlled variables. *Ind. Eng. Chem.*, 42, 3273–3284.

[Skogestad & Postlethwaite, 2005] Skogestad, S. & Postlethwaite, I. (2005). *Multivariable Feedback Control*. John Wiley & Sons, Ltd.

[Syrmos et al., 1997] Syrmos, V., Abdallah, C., Dorato, P., & Grigoriadis, K. (1997). Static Output Feedback: A survey. *Automatica*, 33, 125–137.

[Wang, 1996] Wang, X. A. (1996). Grassmannian, Central Projection, and Output Feedback Pole Assignment of Linear Systems. *IEEE Trans. Automatic Control*, 41, 786–794.

[Wiener, 1948] Wiener, N. (1948). *Cybernetics: or Control and Communication in the Animal and the Machine*. New York: Wiley.

[Ziegler & Nichols, 1942] Ziegler, J. & Nichols, N. (1942). Optimum setting for PID controllers. *Transactions of ASME*, 64, 759–768.

Appendix A

Linear system theory

This appendix gathers some basic linear system theory that the author did not see fit in the main text.

A.1 The generalized plant

The *generalized plant* is a general control problem formulation in which any control problem can be formulated. The general configuration is depicted in Figure A.1 below.

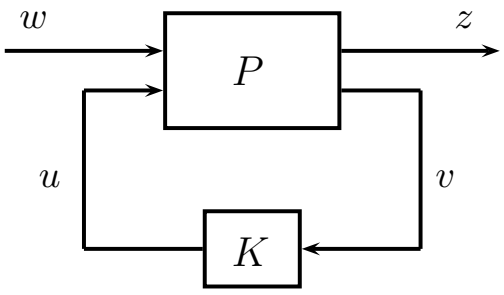


Figure A.1: General control configuration

The general configuration shown in A.1 is described by

$$\begin{bmatrix} z \\ v \end{bmatrix} = P(s) \begin{bmatrix} w \\ u \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}, \tag{A.1}$$

$$u = K(s)v, \tag{A.2}$$

with a state-space realization of the the generalized plant P given by

$$P \stackrel{s}{=} \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right]. \tag{A.3}$$

In this setting u are the control variables, v the measured variables, w the exogenous signals such as disturbances and reference signals, and z are the so-called "error" signals which are to be minimized in some sense to meet the control objectives. The closed-loop transfer function from w to z is given by the linear fractional transform

$$z = F_l(P,K)w. \tag{A.4}$$

A.2 Linear fractional transformations

Linear fractional transformations (LFTs) are a useful tool in many control analysis and design problems. Consider a matrix of dimension $(n_1 + n_2) \times (m_1 + m_2)$ partitioned as follows:

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}. \tag{A.5}$$

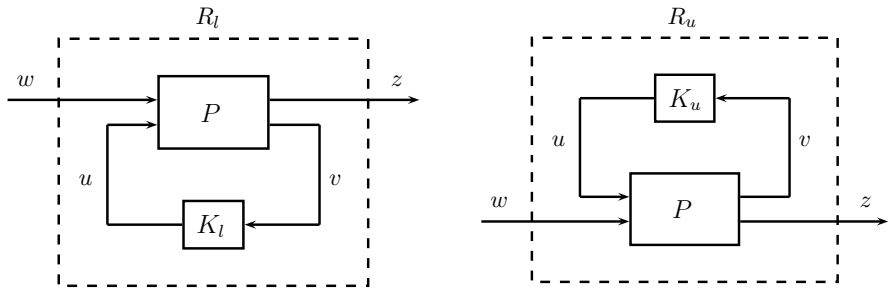
Let the matrices K_l and K_u have dimensions $m_2 \times n_2$ and $m_1 \times n_1$, respectively. K_l and K_u are then compatible with the lower and upper partition of P, respectively. Adopting the notation used by [Skogestad & Postlethwaite, 2005] we define the lower and upper LFT as:

$$F_l(P,K_l) \triangleq P_{11} + P_{12}K_l(I - P_{22}K_l)^{-1}P_{21}, \tag{A.6}$$

$$F_u(P,K_u) \triangleq P_{22} + P_{21}K_l(I - P_{11}K_l)^{-1}P_{12}. \tag{A.7}$$

The lower fractional transformation $F_l(P,K_l)$ is the transfer function R_l from w to z in Figure A.2a. $F_l(P,K_l)$ results from wrapping feedback K_l around the lower part of P.

$$z = P_{11}w + P_{12}u, \quad v = P_{21}w + P_{22}u, \quad u = Kv. \tag{A.8}$$



(a) R_l as lower LFT in terms of K_l (b) R_u as upper LFT in terms of K_u

Figure A.2: Illustration of lower and upper LFT

Eliminating u and v from these relations yields

$$z = R_l w = F_l(P, K_l) w = \left[P_{11} + P_{12} K_l (I - P_{22} K_l)^{-1} P_{21} \right] w, \tag{A.9}$$

where R_l is written as a lower LFT of P in terms of K_l. Similarly, we obtain the upper LFT, $R_u = F_u(P, K_u)$, by wrapping feedback K_u around the upper part of P, as illustrated in Figure A.2b. Note the important property that an interconnection of LFTs results in a LFT [Skogestad & Postlethwaite, 2005].

Appendix B

Derivations

This appendix holds several of the derivations used in the mathematical framework of this thesis.

B.1 Truncation of prediction horizon

To enforce optimality on the infinite horizon with a finite number of optimization variables we employ the usual trick of truncating the prediction horizon (referred to as *horizon* hereafter). This section shows how to truncate or split the horizon of a QP problem by assuming a given control law after the horizon. When constraints are added to the QP problem the finite-horizon formulation becomes an approximation of the infinite-horizon problem. This is because the constraints only are considered a finite number of steps ahead. However, the approximation becomes better as the prediction horizon increases. We will first consider the discrete-time QP problem, then we treat the continuous case.

B.1.1 Discrete-time case

The discrete-time process model is

$$x_{k+1} = Ax_k + Bu_k \tag{B.1}$$

and the infinite-horizon objective function is

$$J = \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right). \tag{B.2}$$

We start by splitting our infinite sum into two parts. The horizon is denoted N.[1]

$$J = \sum_{k=N}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right). \tag{B.3}$$

We wish to optimize over the infinite horizon with a finite number of optimization variables. That is, we seek a matrix P so that

$$\sum_{k=N}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) = x_N^T P x_N. \tag{B.4}$$

This enables us to write the infinite horizon problem as an finite horizon problem

$$J = \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) = \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + x_N^T P x_N, \tag{B.5}$$

where P is dependent on which control scheme we apply after the horizon. If we choose $u_k = -Kx_k$, $\forall k \geq N$, where K is the LQR controller, P solves the Ricatti equation as we will see. Denoting $\Gamma = (A - BK)$ and $\Psi = Q + K^\top R K$ we get

$$x_{k+1} = \Gamma x_k, \quad \forall k \geq N, \tag{B.6}$$

and the sum becomes

$$\sum_{k=N}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right) = x_N^\top \left(\Psi + \Gamma^\top \Psi \Gamma + (\Gamma^\top)^2 \Psi \Gamma^2 + \ldots \right) x_N, \tag{B.7}$$

which converge as long as Γ has eigenvalues inside the unit disc. This holds if K is the LQ optimal controller, under stabilizability of (A, B). We now derive the expression for P by observing that

$$\sum_{k=N}^{\infty} \left(x_k^\top \Psi x_k \right) = x_N^T P x_N, \tag{B.8}$$

and by starting at $k = N + 1$ we get

$$\sum_{k=N+1}^{\infty} \left(x_k^\top \Psi x_k \right) = x_{N+1}^T P x_{N+1} = x_N^T \Gamma^\top P \Gamma x_N. \tag{B.9}$$

[1] With MPC we often distinguish between the *control horizon* and *prediction horizon*. Here we have assumed that the control horizon equals the prediction horizon for simplicity.

We have that

$$\sum_{k=N}^{\infty}\left(x_k^\top \Psi x_k\right) = \sum_{k=N+1}^{\infty}\left(x_k^\top \Psi x_k\right) + x_N^T \Psi x_N \tag{B.10}$$

$$\Rightarrow x_N^T P x_N - x_N^T \Psi x_N = \sum_{k=N+1}^{\infty}\left(x_k^\top \Psi x_k\right) = x_N^T \Gamma^\top P \Gamma x_N \tag{B.11}$$

which means that we must have

$$P - \Psi = \Gamma^\top P \Gamma \tag{B.12}$$

$$\Rightarrow P = (A - BK)^\top P (A - BK) + Q + K^\top R K. \tag{B.13}$$

Eq. (B.13) becomes the well-known discrete-time algebraic Ricatti equation (DARE) when K is the LQ optimal controller (i.e. $K = (R + B^\top P B)^{-1} B^\top P A$). We further note that if u_k is set to zero after the control horizon we get the discrete Lyapunov equation

$$P = A^\top P A + Q. \tag{B.14}$$

Summing up we see that if LQR control is desired after the control horizon we simply choose P as the solution of the Ricatti equation (B.13), which already is calculated when finding the LQ optimal controller K. If control is set to zero we have to solve (B.14). Note that assuming zero control action only make sense for open-loop stable systems, otherwise the sum in (B.7) will not converge.

B.1.2 Continuous-time case

We now consider the continuous case with the process model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{B.15}$$

and the infinite-horizon QP problem

$$J = \int_0^\infty x(t)^\top Q x(t) + u(t)^\top R u(t)\, dt \tag{B.16}$$

$$= \int_{\bar{t}}^\infty x(t)^\top Q x(t) + u(t)^\top R u(t)\, dt + \int_0^{\bar{t}} x(t)^\top Q x(t) + u(t)^\top R u(t)\, dt, \tag{B.17}$$

where the horizon is denoted $\bar{t}$. Again we choose the control law $u(t) = -Kx(t)$ and denote $\Psi = Q + K^\top R K$ and $\Gamma = (A - BK)$. As before we wish to find the matrix P so that

$$\int_{\bar{t}}^\infty x(t)^\top \Psi x(t)\, dt = \left(x(t)^\top P x(t)\right)_{t=\bar{t}}. \tag{B.18}$$

The integral in Eq. (B.18) converge if $x(t) \to 0$ as $t \to \infty$. Hence, we must require that all eigenvalues of Γ lies in the left half-plane. This holds for the LQ optimal controller, under stabilizability of (A, B). Differentiation of Eq. (B.18) gives

$$\left[x(t)^\top \Psi x(t)\right]_{\bar{t}}^{\infty} = \frac{\mathrm{d}}{\mathrm{d}t}\left(x(t)^\top P x(t)\right)_{t=\bar{t}}. \tag{B.19}$$

Under the assumption of an asymptotically stable closed-loop system we have that

$$\left[x(t)^\top \Psi x(t)\right]_{t=\bar{t}}^{\infty} = -x(\bar{t})^\top \Psi x(\bar{t}). \tag{B.20}$$

Further we have that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(x(t)^\top P x(t)\right)_{t=\bar{t}} = \dot{x}(\bar{t})^\top P x(\bar{t}) + x(\bar{t})^\top P \dot{x}(\bar{t}) \tag{B.21}$$

$$= x(\bar{t})^\top \Gamma^\top P x(\bar{t}) + x(\bar{t})^\top P \Gamma x(\bar{t}). \tag{B.22}$$

Substituting into (B.19) and rearranging gives

$$x(\bar{t})^\top \left(\Gamma^\top P + P\Gamma + \Psi\right) x(\bar{t}) = 0 \tag{B.23}$$

and we obtain

$$(A - BK)^\top P + P(A - BK) + Q + K^\top R K = 0, \tag{B.24}$$

which becomes the well-known continuous-time algebraic Ricatti equation (CARE) when K is the LQ optimal controller, i.e. $K = R^{-1}B^\top P$. With zero control action after the horizon, that is $u(t) = 0$, (B.24) simplifies to the continuous Lyapunov equation

$$A^\top P + PA + Q = 0. \tag{B.25}$$

As in the discrete-time case, assuming zero control action after the horizon only make sense for open-loop stable systems.

B.1.3 Static output feedback case

With static output feedback we have that $y = Cx$ (assuming $D = 0$), $u = -K^y y$, and $\Gamma = (A - BK^y C)$. Following the derivations presented above it is rather easy to show that Eqs. (B.13) and (B.24) become

$$P = (A - BK^y C)^\top P (A - BK^y C) + Q + (K^y C)^\top R (K^y C) \quad \text{(Disc. case)} \tag{B.26}$$

$$0 = (A - BK^y C)^\top P + P(A - BK^y C) + Q + (K^y C)^\top R (K^y C) \quad \text{(Cont. case)} \tag{B.27}$$

with static output feedback. These equations does however require that the system is output stabilizable and that the static feedback law implemented after the horizon stabilizes the system.

B.2 Deriving J_{uu} and J_{ud}

In [Manum et al., 2007] the second-order derivatives of the infinite horizon open-loop objective function are derived. The expressions for the second-order derivatives J_{uu} and J_{ud} are used extensively in this thesis and the derivation is repeated here for validation and completeness.

Given the stable process model (A has all eigenvalues within the units circle)

$$x_{k+1} = Ax_k + Bu_k \tag{B.28}$$

with x_0 given, and consider the *open-loop* optimization problem:

$$\min_u J = \sum_{k=0}^{\infty} \left(x_k^\top Q x_k + u_k^\top R u_k \right), \tag{B.29}$$

where the N first inputs are gathered in the optimization vector $u = [u_0, u_1, \ldots, u_{N-1}]^\top$. We now rewrite J by splitting the infinite horizon into two parts and setting the prediction horizon equal to the control horizon, i.e.

$$\begin{aligned} J &= \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + \sum_{k=N}^{\infty} \left(x_k^\top Q x_k \right) \\ &= \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right) + x_N^\top P x_N \\ &= x_0^\top Q x_0 + \sum_{k=1}^{N-1} \left(x_k^\top Q x_k \right) + \sum_{k=0}^{N-1} \left(u_k^\top R u_k \right) + x_N^\top P x_N. \end{aligned} \tag{B.30}$$

Here, P solves discrete Lyapunov equation $P = A^\top P A + Q$, which means that control is switched off after the horizon N (as discussed in B.1). This is reasonable since A is assumed stable. From the process model we derive the following relationship:

$$x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} \tag{B.31}$$

$$\begin{aligned} &\Downarrow \\ x_1 &= Ax_0 + Bu_0 \\ x_2 &= A^2 x_0 + ABu_0 + Bu_1 \\ &\vdots \\ x_{N-1} &= A^{N-1} x_0 + A^{N-2} B u_0 + \ldots + B u_{N-2} \end{aligned} \tag{B.32}$$

Assembling the future states in a vector $x = [x_1, x_2, \ldots, x_{N-1}]^\top$ enables us to write the predictions on the compact form

$$x = \underbrace{\begin{bmatrix} A \\ A^2 \\ A^3 \\ \vdots \\ A^{N-1} \end{bmatrix}}_{a} x_0 + \underbrace{\begin{bmatrix} B & 0 & 0 & \ldots & 0 & 0 \\ AB & B & 0 & \ldots & 0 & 0 \\ A^2B & AB & B & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ A^{N-2}B & A^{N-3}B & \ldots & AB & B & 0 \end{bmatrix}}_{\hat{A}} u. \tag{B.33}$$

To simplify the notation we assume that the weight matrices Q and R is diagonal and constant throughout the horizon, and we write

$$\hat{Q} = \text{diag}(Q, Q, \ldots, Q), \tag{B.34}$$

$$\hat{R} = \text{diag}(R, R, \ldots, R). \tag{B.35}$$

Further we have that

$$\begin{aligned} \sum_{k=1}^{N-1} x_k^\top Q x_k = x^\top \hat{Q} x &= (ax_0 + \hat{A}u)^\top \hat{Q}(ax_0 + \hat{A}u) \\ &= x_0^\top a^\top \hat{Q} a x_0 + 2u^\top \hat{A}^\top \hat{Q} a x_0 + u^\top \hat{A}^\top \hat{Q} \hat{A} u \end{aligned} \tag{B.36}$$

and

$$\sum_{k=0}^{N-1} u_k^\top R u_k = u^\top \hat{R} u. \tag{B.37}$$

Using (B.31) we get that the final state is

$$\begin{aligned} x_N &= A^N x_0 + A^{N-1} B u_0 + \ldots + B u_{N-1} \\ &= A^N x_0 + \underbrace{\begin{bmatrix} A^{N-1}B & A^{N-2}B & \ldots & AB & B \end{bmatrix}}_{\Lambda} u \end{aligned} \tag{B.38}$$

and the term $x_N^\top P x_N$ can be written

$$\begin{aligned} x_N^\top P x_N &= (A^N x_0 + \Lambda u)^\top P (A^N x_0 + \Lambda u) \\ &= x_0^\top (A^N)^\top P A^N x_0 + 2 x_0^\top (A^N)^\top P \Lambda u + u^\top \Lambda^\top P \Lambda u. \end{aligned} \tag{B.39}$$

Comparing (B.36), (B.37), and (B.39) with (B.30) we easily see that

$$J_{uu} = 2\left(\hat{A}^\top \hat{Q} \hat{A} + \Lambda^\top P \Lambda + \hat{R}\right), \tag{B.40}$$

$$J_{ud} = 2\left(\hat{A}^\top \hat{Q} a + \Lambda^\top P A^N\right). \tag{B.41}$$

We now focus on calculating the J_{uu} matrix. We observe that $\hat{A}$ and Λ can be partitioned into $\hat{A} = \tilde{A}\tilde{B}$ and $\Lambda = \tilde{\Lambda}\tilde{B}$, where $\tilde{B} = \text{diag}(B, B, \ldots, B)$ with a correct dimension. This allows us to write:

$$(\hat{A}^\top \hat{Q})\hat{A} = \tilde{B}^\top (\tilde{A}^\top \hat{Q})\tilde{A}\tilde{B} \tag{B.42}$$

$$= \tilde{B}^\top \underbrace{\begin{bmatrix} Q & A^\top Q & (A^2)^\top Q & \ldots & (A^{N-2})^\top Q \\ 0 & Q & A^\top Q & \ldots & (A^{N-3})^\top Q \\ 0 & 0 & Q & \ldots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & Q \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix}}_{\tilde{A}^\top Q} \underbrace{\begin{bmatrix} I & 0 & 0 & \ldots & 0 & 0 \\ A & I & 0 & \ldots & 0 & 0 \\ A^2 & A & I & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ A^{N-2} & A^{N-3} & \ldots & A & I & 0 \end{bmatrix}}_{\tilde{A}} \tilde{B}. \tag{B.43}$$

We proceed by finding

$$\Lambda^\top P \Lambda = \begin{bmatrix} B^\top (A^{N-1})^\top P \\ B^\top (A^{N-2})^\top P \\ \vdots \\ B^\top A^\top P \\ B^\top P \end{bmatrix} \begin{bmatrix} A^{N-1}B & A^{N-2}B & \ldots & AB & B \end{bmatrix}. \tag{B.44}$$

The first block element of J_{uu}, lets denote it $J_{uu,11}$, is then

$$\begin{aligned} J_{uu,11} &= B^\top \left[Q + A^\top Q A + (A^2)^\top Q A^2 + \ldots + (A^{N-2})^\top Q A^{N-2} \right] B \\ &\quad + B^\top (A^{N-1})^\top P A^{N-1} B + R \qquad \text{(B.45)} \\ &= B^\top \left(\sum_{i=0}^{N-2} \left[(A^i)^\top (P - A^\top P A) A^i \right] + (A^{N-1})^\top P A^{N-1} \right) B + R \\ &= B^\top \left(\sum_{i=0}^{N-1} (A^i)^\top P A^i - \sum_{i=1}^{N-1} (A^i)^\top P A^i \right) B + R \\ &= B^\top P B + R, \qquad \text{(B.46)} \end{aligned}$$

where we have used the relation $P = A^\top P A + Q$ to substitute for Q. Continuing with the next elements we observe a pattern on the upper and lower triangular part of the matrix. The result is:

$$J_{uu} = 2 \begin{bmatrix} B^\top P B + R & B^\top A^\top P B & \ldots & B^\top (A^{N-1})^\top P B \\ B^\top P A B & B^\top P B + R & \ldots & B^\top (A^{N-2})^\top P B \\ \vdots & \vdots & \ddots & \vdots \\ B^\top P A^{N-1} B & B^\top P A^{N-2} B & \ldots & B^\top P B + R \end{bmatrix}. \tag{B.47}$$

We now try to find J_{ud} and we start by expanding the expression for $\hat{A}^\top \hat{Q} a$ in Eq. (B.41).

$$\hat{A}^\top \hat{Q} a = \tilde{B}^\top (\tilde{A}^\top \hat{Q}) a \tag{B.48}$$

$$= \tilde{B}^\top \begin{bmatrix} Q & A^\top Q & (A^2)^\top Q & \dots & (A^{N-2})^\top Q \\ 0 & Q & A^\top Q & \dots & (A^{N-3})^\top Q \\ 0 & 0 & Q & \dots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & Q \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N-1} \end{bmatrix} \tag{B.49}$$

$$= \tilde{B}^\top \begin{bmatrix} QA + A^\top QA^2 + \dots + (A^{N-2})^\top QA^{N-1} \\ QA^2 + A^\top QA^3 + \dots + (A^{N-3})^\top QA^{N-1} \\ \vdots \\ QA^{N-1} \\ 0 \end{bmatrix} \tag{B.50}$$

$$= \tilde{B}^\top \begin{bmatrix} PA - (A^{N-1})^\top PA^N \\ PA^2 - (A^{N-2})^\top PA^N \\ \vdots \\ PA^{N-1} - A^\top PA^N \\ 0 \end{bmatrix}, \tag{B.51}$$

where we again have utilized $P = A^\top PA + Q$ to substitute for Q. Continuing we have that

$$(A^N)^\top P\Lambda = \Lambda^\top PA^N = \tilde{B}^\top \tilde{\Lambda}^\top PA^N = \tilde{B}^\top \begin{bmatrix} (A^{N-1})^\top PA^N \\ (A^{N-2})^\top PA^N \\ \vdots \\ A^\top PA^N \\ PA^N \end{bmatrix}. \tag{B.52}$$

Finally, summation of (B.51) and (B.52) yields the result:

$$J_{ud} = 2 \begin{bmatrix} B^\top P \\ B^\top PA \\ \vdots \\ B^\top PA^{N-1} \end{bmatrix} A. \tag{B.53}$$

B.3 Approximation of the $\mathscr{H}_2$-optimal static output feedback problem

In Chapter 2.3.9 we saw how the $\mathscr{H}_2$ problem can be formulated to represent the deterministic initial value problem (LQR problem). An interpretation of this is that the $\mathscr{H}_2$ norm may be viewed as the 2-norm of the impulse response resulting from applying unit impulses to each initial state. Consider the LTI system

$$x_{k+1} = Ax_k + Bu_k, \tag{B.54}$$

$$y_k = Cx_k, \tag{B.55}$$

we then have that

$$J = E\left\{\sum_{i=0}^{\infty} x_i^\top Qx_i + u_i^\top Ru_i\right\} = ||F_l(P,K)||_2^2, \tag{B.56}$$

when the generalized plant P is defined as in Chapter 2.3.9.

With the above knowledge we now present an approximation of the $\mathscr{H}_2$ problem in the deterministic case (without stochastic signals). By adding the system constraints in Eqs. (B.54) and (B.55), and the constraint $u_k = -Ky_k$ to the cost function above we get the optimization problem:

$$\min_K J = \sum_{i=0}^{\infty} x_i^\top Qx_i + u_i^\top Ru_i \tag{B.57}$$

$$\text{s.t. } x_{i+1} = (A - BKC)x_i. \tag{B.58}$$

From the theory in Chapter 2.3 we know that the solution to the LQR problem lies on the constraint $u_k = -Kx_k$, that is, the solution is a static state feedback. Thus, when $C = I$ the above optimization problem gives the same solution as the LQR problem. When $C \neq I$ we no longer have the equivalence and we have to solve the above optimization problem to find the $\mathscr{H}_2$-optimal controller (static output feedback).

The infinite-horizon problem can be approximated by a finite-horizon problem of length N, i.e.

$$\min_K J = x_N^\top Px_N + \sum_{i=0}^{N-1} x_i^\top Qx_i + u_i^\top Ru_i \tag{B.59}$$

$$\text{s.t. } x_{i+1} = (A - BKC)x_i. \tag{B.60}$$

From the derivations in Appendix B.1 we know that the relation between the finite-horizon and infinite-horizon problem is exact when the terminal state weight matrix P solves

$$P = (A - BKC)^\top P(A - BKC) + Q + (KC)^\top R(KC). \tag{B.61}$$

To remove the unwanted dependance of P on K we choose $K = 0$ in the above equation, meaning that we turn off control (i.e. $u = 0$) after the horizon. P is then the solution of the discrete Lyapunov equation $P = A^\top PA + Q$. This approximation of the infinite-horizon problem gives a suboptimal solution, but as the horizon length is increased the approximation is improved. When $N \to \infty$ we retrieve the infinite-horizon problem.

Using the relation $x_i = (A - BKC)^i x_0$ we get by substitution that

$$J = x_0^\top M(K) x_0, \tag{B.62}$$

where, by defining $A_c = (A - BKC)$,

$$M(K) = (A_c^N)^\top P A_c^N + \sum_{i=0}^{N-1} (A_c^i)^\top (Q + (KC)^\top R(KC)) A_c^i. \tag{B.63}$$

The impulse-response objective function is then

$$J_{ir} = \sum_{i=1}^{n_x} e_i^\top M(K) e_i = \text{trace}(M(K)), \tag{B.64}$$

where e_i is a vector of length n_x with a 1 in it's i'th place. To summarize we have that the alternative closed-loop formulation

$$\min_K \text{trace}(M(K)) \tag{B.65}$$

is the same as minimizing the $||F_l(P,K)||_2^2$ when $N \to \infty$. We further have that minimizing $||F_l(P,K)||_2$ is the same as minimizing $\sqrt{\text{trace}(M(K))}$.

B.4 Model augmentation

This section shows how a continuous-time state-space model can be augmented with the dynamics of a MIMO PID controller. Consider the following MIMO PID controller on parallel form with limited derivative action

$$u(s) = (K_p + K_i \frac{1}{s} + K_d \frac{s}{\epsilon s + 1}) e(s) \tag{B.66}$$

$$= K_p y^p + K_i y^i + K_d y^d, \tag{B.67}$$

where $y^p = e$, $y^i = \frac{1}{s} e$, $y^d = \frac{s}{\epsilon s+1} e$, and $e = r - y$. The new variables represent e, $\int e\,dt$, and $\dot{e}$, respectively. Please note that y^d is an approximation of $\dot{e}$ since a filter is included. K_p, K_i, and K_d are $n_u \times n_y$ matrices.

We now want to augment the linear system

$$\dot{x} = Ax + Bu, \tag{B.68}$$

$$y = Cx + Du, \tag{B.69}$$

with the controller states and choose the desired outputs:

$$\begin{bmatrix} y^p \\ y^i \\ y^d \end{bmatrix} = \begin{bmatrix} e \\ \frac{1}{s} e \\ \frac{s}{\epsilon s+1} e \end{bmatrix}. \tag{B.70}$$

The integral action is included through the state σ, that is

$$y^i = \sigma = \int e\,dt \Rightarrow \dot{\sigma} = e = r - (Cx + Du), \tag{B.71}$$

where we have used (B.69) to substitute for y. To include limited derivative action we study the expression

$$y^d = \frac{s}{\epsilon s+1} e = \frac{s}{\epsilon s+1} (r - Cx - Du) \tag{B.72}$$

$$= -\frac{Cs}{\epsilon s+1} x - \frac{Ds}{\epsilon s+1} u + \frac{s}{\epsilon s+1} r \tag{B.73}$$

$$= -\frac{C(Ax+Bu)}{\epsilon s+1} - \frac{Ds}{\epsilon s+1} u + \frac{s}{\epsilon s+1} r, \tag{B.74}$$

where we have used the relation $s \cdot x = Ax + Bu$ from (B.68). The last two terms in (B.74) can be written

$$D \frac{s}{\epsilon s+1} u = \frac{1}{\epsilon} D (1 - \frac{1}{\epsilon s+1}) u \tag{B.75}$$

$$\frac{s}{\epsilon s+1} r = \frac{1}{\epsilon} (1 - \frac{1}{\epsilon s+1}) r, \tag{B.76}$$

which gives

$$y^d = -\frac{C(Ax+Bu)}{\epsilon s+1} - \frac{1}{\epsilon}D(1-\frac{1}{\epsilon s+1})u + \frac{1}{\epsilon}(1-\frac{1}{\epsilon s+1})r \tag{B.77}$$

$$= -\frac{1}{\epsilon}\left(\frac{\epsilon CAx + (\epsilon CB - D)u + r}{\epsilon s+1}\right) - \frac{1}{\epsilon}Du + \frac{1}{\epsilon}r \tag{B.78}$$

$$= -\frac{1}{\epsilon}\delta - \frac{1}{\epsilon}Du + \frac{1}{\epsilon}r, \tag{B.79}$$

where we have defined the new state δ. Continuing we have that

$$\delta = \left(\frac{\epsilon CAx + (\epsilon CB - D)u + r}{\epsilon s+1}\right) \tag{B.80}$$

$$(\epsilon s+1)\delta = (\epsilon CAx + (\epsilon CB - D)u + r) \tag{B.81}$$

and moving over to the time domain gives

$$\dot{\delta} = -\frac{1}{\epsilon}\delta + \frac{1}{\epsilon}(\epsilon CAx + (\epsilon CB - D)u + r) \tag{B.82}$$

$$= CAx - \frac{1}{\epsilon}\delta + \frac{1}{\epsilon}(\epsilon CB - D)u + \frac{1}{\epsilon}r. \tag{B.83}$$

Augmenting the system with σ and δ, and chosing y^p, y^i, and y^d as outputs gives

$$\begin{bmatrix}\dot{x}\\ \dot{\sigma}\\ \dot{\delta}\end{bmatrix} = \begin{bmatrix}A & 0 & 0\\ -C & 0 & 0\\ CA & 0 & -\frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}x\\ \sigma\\ \delta\end{bmatrix} + \begin{bmatrix}B & 0\\ -D & I\\ \frac{1}{\epsilon}(\epsilon CB - D) & \frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix} \tag{B.84}$$

$$\begin{bmatrix}y^p\\ y^i\\ y^d\end{bmatrix} = \begin{bmatrix}-C & 0 & 0\\ 0 & I & 0\\ 0 & 0 & -\frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}x\\ \sigma\\ \delta\end{bmatrix} + \begin{bmatrix}-D & I\\ 0 & 0\\ -\frac{1}{\epsilon}D & \frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix}, \tag{B.85}$$

where r is regarded as a new input. The augmented system has eigenvalues at $\lambda = \{\text{eig}(A), 0, \ldots, 0, -\frac{1}{\epsilon}, \ldots, -\frac{1}{\epsilon}\}$, with n_y zeros coming from the integrators and n_y eigenvalues at $-\frac{1}{\epsilon}$ resulting from the filter on the derivative action. PI control is obtained by removing δ and the lower y^d part of (B.85). PD control is obtained by removing σ and y^i from (B.84) and (B.85). Additionally, we can exclude derivative action on the reference signal r by setting the bottom right r-part ($\frac{1}{\epsilon}I$) in Eqs. (B.84) and (B.85) to zero, hence removing it from δ and y^d. To examplify; without derivative action on r and $D = 0$ the system simplifies to

$$\begin{bmatrix}\dot{x}\\ \dot{\sigma}\\ \dot{\delta}\end{bmatrix} = \begin{bmatrix}A & 0 & 0\\ -C & 0 & 0\\ CA & 0 & -\frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}x\\ \sigma\\ \delta\end{bmatrix} + \begin{bmatrix}B & 0\\ 0 & I\\ CB & 0\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix} \tag{B.86}$$

$$\begin{bmatrix}y^p\\ y^i\\ y^d\end{bmatrix} = \begin{bmatrix}-C & 0 & 0\\ 0 & I & 0\\ 0 & 0 & -\frac{1}{\epsilon}I\end{bmatrix}\begin{bmatrix}x\\ \sigma\\ \delta\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}u\\ r\end{bmatrix}. \tag{B.87}$$

With the PID dynamics included in the model the feedback law is $u = [K_p\ K_i\ K_d] \cdot [y^p\ y^i\ y^d]^\top$. When negative feedback is desired we can simply choose the outputs to be $-[y^p\ y^i\ y^d]^\top$ by using the alternative model

$$\begin{bmatrix} y^p \\ y^i \\ y^d \end{bmatrix} = \begin{bmatrix} C & 0 & 0 \\ 0 & -I & 0 \\ 0 & 0 & \frac{1}{\epsilon} I \end{bmatrix} \begin{bmatrix} x \\ \sigma \\ \delta \end{bmatrix} + \begin{bmatrix} D & -I \\ 0 & 0 \\ \frac{1}{\epsilon} D & -\frac{1}{\epsilon} I \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix}, \tag{B.88}$$

where the states are the same as in Eq. (B.84).

Some notes are appropriate at this point. First of all we observe that the system is augmented with $2 \cdot n_y$ states, i.e. the PID controller is of order $2 \cdot n_y$. When $2 \cdot n_y < n_x$ the system with controller can be interpereted as a static output feedback problem (as discussed in section 2.1). Further, the PID controller with limited derivative action approximates the derivative of y by filtering the output through a low-pass filter. This means that it never can be as optimal as the LQR, where all states available, even with an order $2 \cdot n_y \geq n_x$.

B.4.1 Model augmentation - special cases

Next we will consider a few other cases, starting with a MIMO PI controller with low-pass filtered output. The controller is on the form:

$$u = \left(K_p + K_i \frac{1}{s}\right)\left(\frac{1}{\tau s + 1}\right) y = K_p y^{p,f} + K_i y^{i,f}, \tag{B.89}$$

with $y^{p,f} = \frac{1}{\tau s+1} y$ and $y^{i,f} = \frac{1}{s(\tau s+1)}$. As before K_p and K_i are $n_u \times n_y$ matrices.

We augment the state-space model with the following controller states:

$$\zeta_1 = \frac{1}{\tau s + 1} y \Rightarrow \dot{\zeta}_1 = -\frac{1}{\tau}\zeta_1 + \frac{1}{\tau}(Cx + Du), \tag{B.90}$$

$$\zeta_2 = \frac{1}{s}\zeta_1 \Rightarrow \dot{\zeta}_2 = \zeta_1, \tag{B.91}$$

where we have utilized the relation $y = Cx + Du$. With $y^{p,f}$ and $y^{i,f}$ as outputs the augmented model becomes

$$\begin{bmatrix} \dot{x} \\ \dot{\zeta}_1 \\ \dot{\zeta}_2 \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ C & -\frac{1}{\tau} I & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} x \\ \zeta_1 \\ \zeta_2 \end{bmatrix} + \begin{bmatrix} B \\ D \\ 0 \end{bmatrix} u \tag{B.92}$$

$$\begin{bmatrix} y^{p,f} \\ y^{i,f} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} x \\ \zeta_1 \\ \zeta_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u. \tag{B.93}$$

We now outline how a discrete-time state-space model can be augmented with the discrete version of the MIMO PID controller in (B.67). The proportinal, integrative, and derivative outputs are approximated as follows

$$y_k^p = y_k \text{ (P)}, \quad y_k^i = \frac{T_s}{z-1} y_k \text{ (I)}, \quad y_k^d = \frac{z-1}{T_s z} y_k \text{ (D)}, \tag{B.94}$$

where T_s is the sampling time and z is the time-shift operator. Augmenting the system with y_k^p is trivial as it equals the output y_k. The integrated output y_k^i is introduced by defining the new state

$$\sigma_{k+1} = T_s \sum_{j=0}^{k} y_j = T_s y_k + \sigma_k = T_s(Cx_k + Du_k) + \sigma_k, \tag{B.95}$$

and we have that $y_k^i = \sigma_k$. We proceed by rewriting the expression for the derivative output y_k^d so that

$$y_k^d = \frac{1}{T_s}(y_k - y_{k-1}) = \frac{1}{T_s}(Cx_k + Du_k - y_{k-1}). \tag{B.96}$$

The last term, y_{k-1}, is included through the state δ as follows:

$$\delta_{k+1} = Cx_k + Du_k \Rightarrow \delta_k = Cx_{k-1} + Du_{k-1}, \tag{B.97}$$

which gives

$$y_k^d = \frac{1}{T_s}(Cx_k + Du_k - \delta_k). \tag{B.98}$$

Using the derived expressions for the new outputs y_k^p, y_k^i, and y_k^d we augment the system using the introduced states σ_k and δ_k, yielding:

$$\begin{bmatrix} x_{k+1} \\ \sigma_{k+1} \\ \delta_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ T_s C & 0 & I \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \sigma_k \\ \delta_k \end{bmatrix} + \begin{bmatrix} B \\ T_s D \\ D \end{bmatrix} u_k \tag{B.99}$$

$$\begin{bmatrix} y_k^p \\ y_k^i \\ y_k^d \end{bmatrix} = \begin{bmatrix} C & 0 & 0 \\ 0 & I & 0 \\ \frac{1}{T_s} C & 0 & -\frac{1}{T_s} I \end{bmatrix} \begin{bmatrix} x_k \\ \sigma_k \\ \delta_k \end{bmatrix} + \begin{bmatrix} D \\ 0 \\ \frac{1}{T_s} D \end{bmatrix} u_k. \tag{B.100}$$

Appendix C

Proof of convexity

This appendix is dedicated to the proof of convexity of the impulse-response representation of the $\mathscr{H}_2$-optimal static output feedback problem. In Appendix B.3 we showed that minimizing

$$J = \text{trace}(M(K)) \tag{C.1}$$

is the same as minimizing $||F_l(P,K)||_2^2$ when the horizon length N goes to infinity (in the deterministic case). We also derived that

$$M(K) = (A_c^N)^\top P A_c^N + \sum_{i=0}^{N-1} (A_c^i)^\top (Q + (KC)^\top R(KC)) A_c^i, \tag{C.2}$$

where $A_c = (A - BKC)$ and P solves the discrete Lyapunov equation. To prove that (C.1) is convex in K we will try to show that the Hessian of J is positive semidefinite.[1] We denote the Hessian J_{KK}, where the subscript refers to the second derivative of J with respect to K.

Problem: Show that J is convex with respect to K, i.e. $J_{KK} = \frac{\partial^2 J}{\partial K^2} > 0$, $\forall K \in \mathbb{R}^{n_u \times n_y}$, where n_u and n_y are the number inputs and measurements respectively.

We will study this problem under the following *assumptions*:

1. $Q \geq 0$ and $R > 0$,
2. (A, B, C) is output stabilizable, and

[1] A function is strictly convex if its Hessian is positive definite.

3. $P = P^\top \geq 0$ solves the Lyapunov equation $P = A^\top PA + Q$.

Notice the assumption that (A, B, C) is output stabilizable, which implies that there exists a controller K that makes the closed-loop system $(A - BKC)$ stable (placing all eigenvalues within the unit circle). This assumption cannot be efficiently tested, as mentioned in Chapter 2.1.

C.1 SISO systems with a single state

We first try to prove convexity for SISO systems with a single state, where static output feedback is the same as state feedback. For this proof we change the notation from upper to lower case to underline that we are manipulating scalars. From Assumption 2 we require that $b \neq 0$ and $c \neq 0$. Using the lower case notation we have that

$$J = p(a - bkc)^{2N} + \sum_{i=0}^{N-1} (q + r(kc)^2)(a - bkc)^{2i}. \tag{C.3}$$

We differentiate (C.1) two times with repect to k and obtain the Hessian J_{kk}:

$$\begin{aligned} J_{kk} &= p \cdot 2N(2N-1) \cdot (bc)^2 \cdot (a - bkc)^{2N-2} \\ &\quad + \sum_{i=0}^{N-1} q \cdot 2i(2i-1) \cdot (bc)^2 \cdot (a - bkc)^{2i-2} \\ &\quad + \sum_{i=0}^{N-1} 2r \cdot c^2 \cdot \left[(a - (2i+1)bkc)^2 - (2i+1)i(bkc)^2\right] \cdot (a - bkc)^{2i-2} \qquad \text{(C.4)} \\ &= p \cdot 2N(2N-1) \cdot (bc)^2 \cdot ((a - bkc)^{N-1})^2 \\ &\quad + 2c^2 \sum_{k=0}^{N-1} \left[qb^2(2i-1)i + r(a - (2i+1)bkc)^2 - r(2i+1)i(bkc)^2\right] \cdot ((a - bkc)^{i-1})^2. \end{aligned} \tag{C.5}$$

We observe that the first term is positive if p is positive. From the assumptions we know that p solves the Lyapunov equation and is non-negative. Thus we can eliminate this term from J_{kk} and write

$$J_{kk} \geq 2c^2 \sum_{i=0}^{N-1} \left[qb^2(2i-1)i + r(a - (2i+1)bkc)^2 - r(2i+1)i(bkc)^2\right] \cdot ((a - bkc)^{i-1})^2. \tag{C.6}$$

For $i = 0$ the term inside the sum reduces to $2c^2 \cdot r(a - bkc)^2 \cdot (a - bkc)^{-2} = 2c^2 r$, which clearly is positive. Hence, we remove this term and start the index i at 1. We

now have that

$$J_{kk} > 2c^2 \sum_{i=1}^{N-1} \underbrace{\left[qb^2(2i-1)i + r(a-(2i+1)bkc)^2 - r(2i+1)i(bkc)^2\right]}_{>0?} \cdot \underbrace{((a-bkc)^{i-1})^2}_{>0}. \tag{C.7}$$

The last term inside the brackets poses a problem in showing that the Hessian is positive. We rewrite the terms inside the brackets so that

$$qb^2 \underbrace{(2i^2-i)}_{\geq i} + r(a^2 - 2(2i+1)abkc + \underbrace{(2i^2+3i+1)}_{\geq 5i+1}(bkc)^2) \tag{C.8}$$

$$\geq qb^2 i + r(a^2 - 2(2i+1)abkc + (5i+1)(bkc)^2), \tag{C.9}$$

where we have used the fact that $i^2 \geq i$, $\forall i \in \{1, 2, \ldots\}$. We now collect all terms with the factor i:

$$(qb^2 - 4r(abkc) + 5r(bkc)^2)i + r(a^2 - 2abck + (bkc)^2) \tag{C.10}$$

$$= (qb^2 - 4r(abkc) + 5r(bkc)^2)i + r(a-bkc)^2 \tag{C.11}$$

$$\geq (qb^2 - 4r(abkc) + 5r(bkc)^2)i. \tag{C.12}$$

To delete the term $-4r(abkc)$ we will use the relation $r(a-2bkc)^2 = ra^2 - 4r(abkc) + 4r(bkc)^2$. Substitution yields

$$qb^2 - 4r(abkc) + 5r(bkc)^2 = qb^2 + r(a-bkc)^2 - ra^2 + r(bkc)^2 \tag{C.13}$$

$$\geq qb^2 - ra^2, \tag{C.14}$$

where the index i has been omitted. We now wish to answer for which values of q and r the expression in (C.14) is non-negative, i.e.

$$qb^2 - ra^2 \geq 0. \tag{C.15}$$

All the values in this inequality are positive and we easily see that we must have

$$\frac{q}{r} \geq (\frac{a}{b})^2. \tag{C.16}$$

By requiring (C.16) we have proven that the objective function (C.3) is convex with respect to the controller k. Note that the requirement is not an absolute lower bound, but simply a *sufficient condition* that ensures a convex objective function for SISO systems with one state.

C.2 The general case – Finding an approach

We now consider the general case of MIMO systems with any number of states. Switching back to upper-case notation we have that the objective function is

$$J = \text{trace}\left(A_c^{N\top} P A_c^N + \sum_{i=0}^{N-1} A_c^{i\top} (Q + (KC)^\top R(KC)) A_c^i \right) \tag{C.17}$$

$$\begin{aligned} &= \text{trace}\left((A-BKC)^{N\top} P (A-BKC)^N \right) \\ &\quad + \sum_{i=0}^{N-1} \text{trace}\left((A-BKC)^{i\top} Q (A-BKC)^i \right) \\ &\quad + \sum_{i=0}^{N-1} \text{trace}\left((A-BKC)^{i\top} ((KC)^\top R(KC)) (A-BKC)^i \right). \end{aligned} \tag{C.18}$$

Some comments on the above objective function and the problem of minimizing it are given next:

- J is a scalar function. When K is a vector, J_K is a vector, and J_{KK} is a matrix. When K is a matrix, J_K is a matrix, and J_{KK} becomes a 4-th order tensor.
- The author has tried several approaches in the search for convexity conditions for J, namely: vectorization combined with the kronecker product, elementwise calculation of the derivatives, and diagonalization of the terms in J. None of these gave any results, mainly because they ended up with equations difficult to solve. For example, the idea of the diagonalization approach was to write terms on the form $(A-BKC)^N$ as $Z^{-1}\Lambda^N Z$, where Λ is a matrix with the closed-loop eigenvalues at its diagonal. Then, we get that $\text{trace}\left[((A-BKC)^N)^\top Q (A-BKC)^N\right] = \text{trace}\left[Z^\top \Lambda^N Z^{-\top} Q Z^{-1} \Lambda^N Z\right]$. However, finding Z and Λ requires difficult eigenvalue and eigenvector calculations. Moreover, this formulation clouds the dependance on the parameters in K, rendering the new problem formulation useless.
- In the continuous case the optimal feedback can be found by solving Eqs. (2.10), (2.11), and (2.12). These equations are derived from choosing P as the solution of the Riccati equation (B.27) and differentiating the objective $J = \text{trace}(P)$. Notice that in this case $M(K) = P$, and the finite-horizon formulation is no longer an approximation.
- A promising alternative that the author did not get time to look into is geometric programming [Boyd & Vandenberghe, 2004]. The idea is basically to build the objective function from the bottom up using convex building blocks. This way, the convexity property is maintained from the start.

Appendix D

Simulink schemes

The Simulink diagrams belonging to the examples in Chapter 5 are gathered here.

D.1 Robustness properties of LQG: Simulink schemes

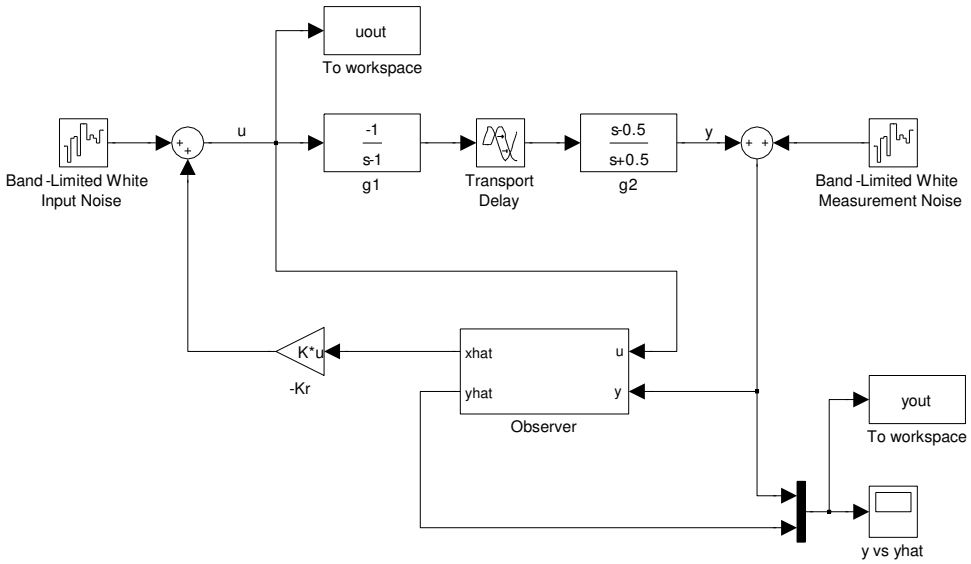


Figure D.1: Simulink scheme of noisy plant and LQG controller.

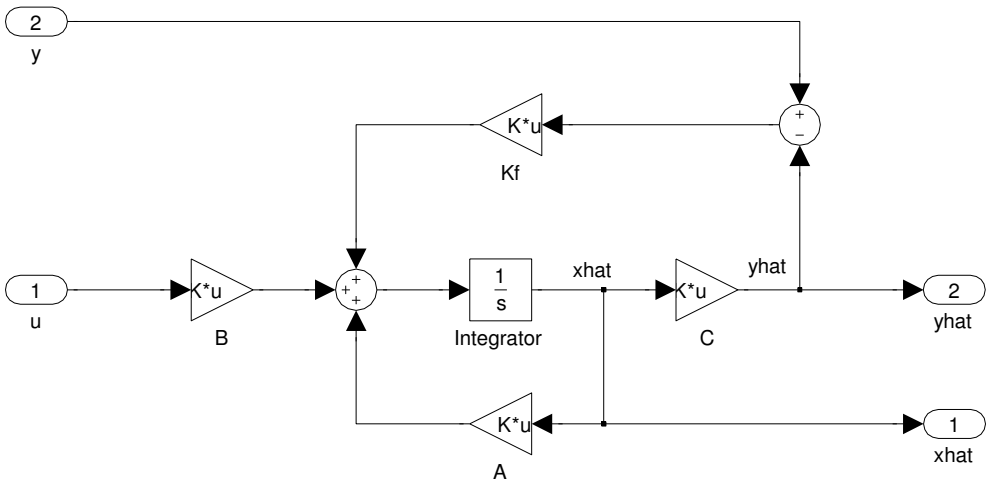


Figure D.2: Observer scheme (continuous steady-state Kalman filter).

D.2 Thermal/Optical Plant: Simulink schemes

The Simulink schemes for the thermal/optical plant example are presented here. Recall that the example studies two plant models for the uDAQ28/LT, one model with 3 inputs and one with 2 inputs. Separate Simulink schemes was used for these models, but the schemes only differ in that the 2-input model contains an input convertion block (from two controller outputs to the three physical plant inputs). Thus, we will only present the Simulink diagrams for the 3-input model.

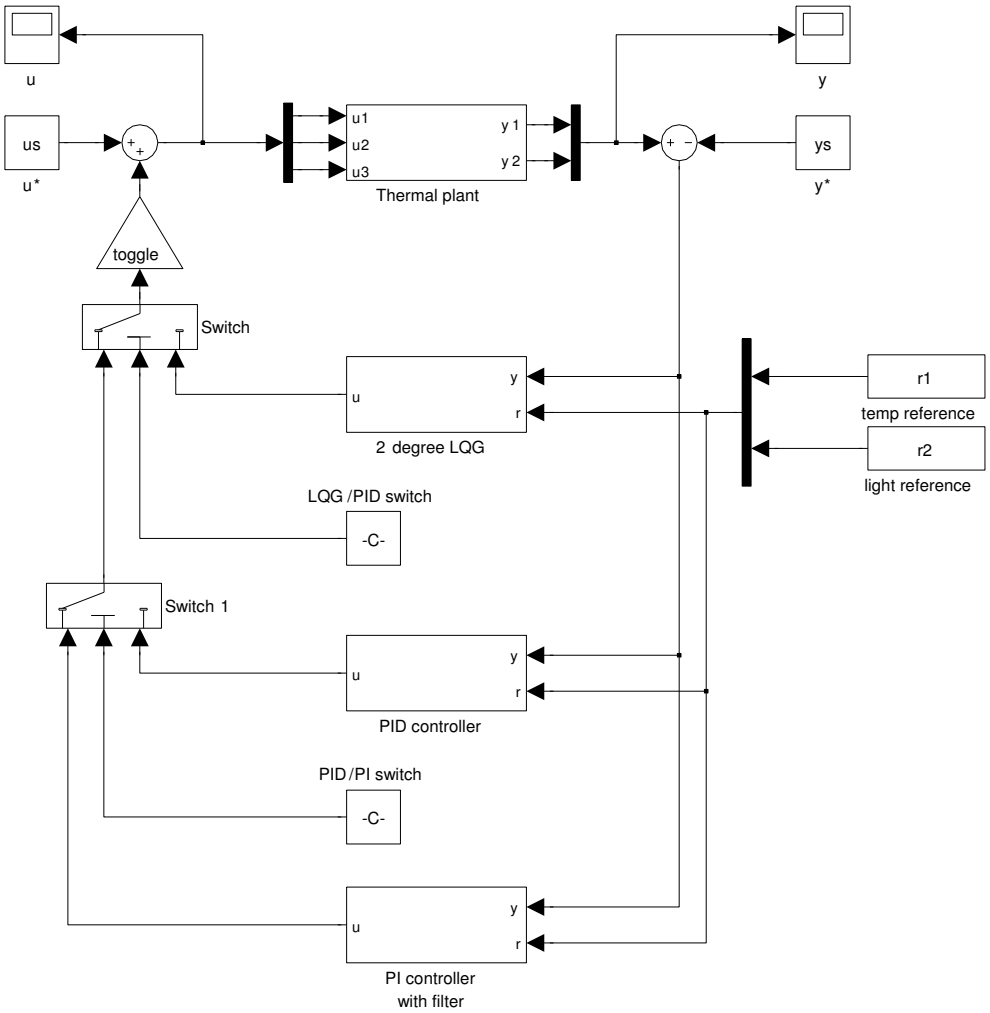


Figure D.3: Simulink scheme of the three controllers and thermal/optical plant. Several switches are included to ease the replacement one controller with another. The reference signal enters at the right-hand side of the scheme.

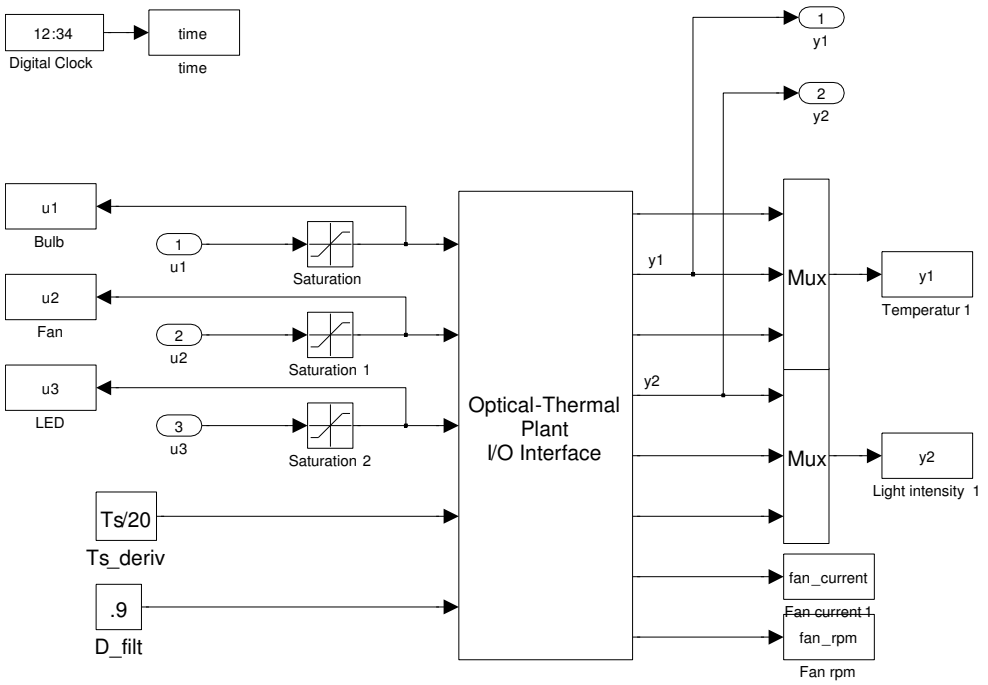


Figure D.4: Simulink scheme showing the internals of the thermal plant subsystem. The "Optical-Thermal Plant I/O Interface" block is responsible for real-time communication with the uDAQ28/LT. Saturation blocks are included to prevent inputs higher than 5 V, which could damage the actuators/electronics.

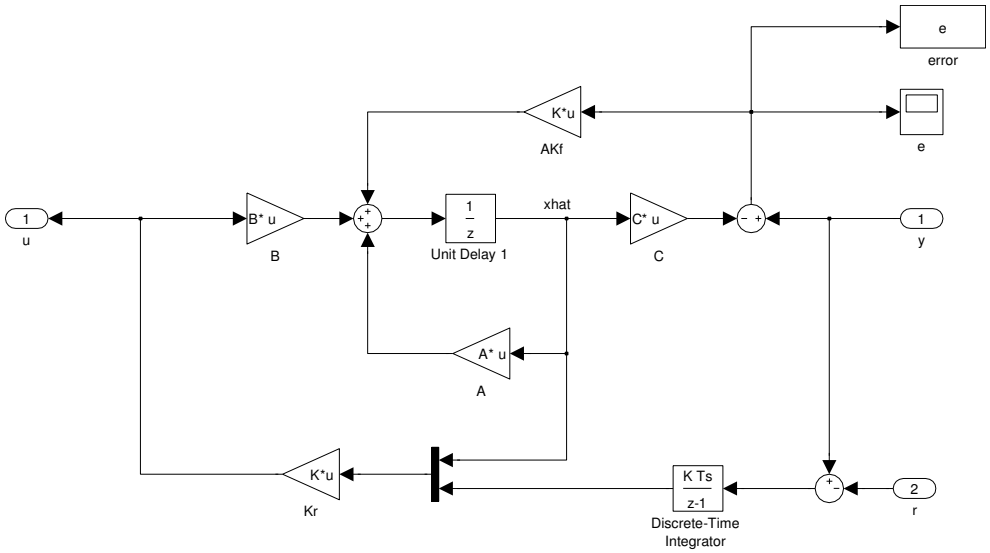


Figure D.5: Simulink scheme of the LQG controller. The controller consists of a discrete steady-state Kalman filter and a LQR.

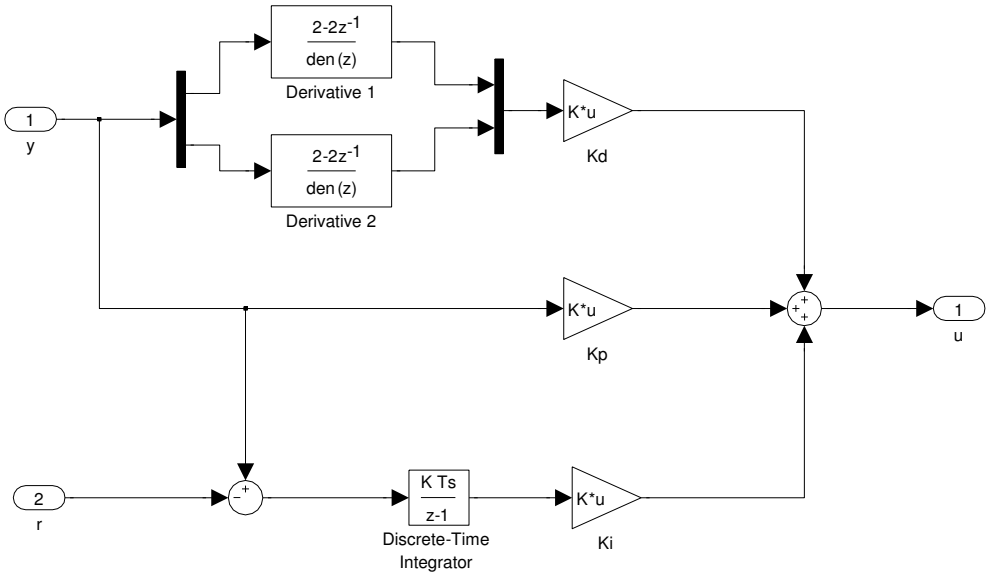


Figure D.6: Simulink scheme of the PID controller.

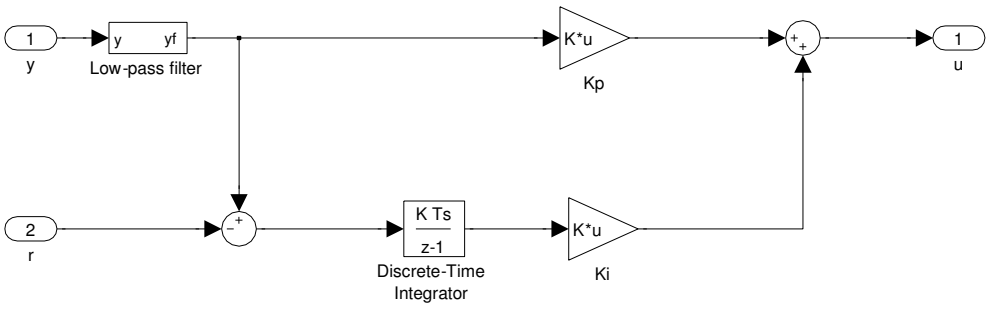


Figure D.7: Simulink scheme of the low-pass filtered PI controller.