Laguerre Bases for Youla-Parametrized Optimal-Controller Design: Numerical Issues and Solutions

Olle Kjellqvist

LUND UNIVERSITY

Department of Automatic Control

MSc Thesis
TFRT-6061
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2018

Abstract

This thesis concerns the evaluation of cost functionals on $\mathcal{H}_2$ when designing optimal controllers using finite Youla parameterizations and convex optimization. We propose to compute inner products of stable, strictly proper systems via solving Sylvester equations. The properties of different state space realizations of Laguerre filters, when performing Ritz expansions of the optimal controller are discussed, and a closed form expression of the output orthogonal realization is presented. An algorithm to exploit Toeplitz substructure when solving Lyapunov equations is discussed, and a method to extend SISO results to MIMO systems using the vectorization operator is proposed. Finally the methods are evaluated on example systems from the industry, where it is shown that properly selecting the cutoff frequency of the filters is an important problem that should be discussed when Laguerre bases are used to parametrize the optimal controller.

Acknowledgements

I would like to thank my supervisor, Olof Troeng, for his sharp intellect, fabulous drive and over all pleasant characteristics. My assistant supervisor Pontus Giselsson who assisted by sharing his vast knowledge of convex optimization. The feedback from Bo Bernhardsson and Anton Cervin has been most helpful in where to focus my attention. This thesis owes much of its readability to Leif Andersson, who provided valuable feedback on typesetting. This thesis owes its existence to everybody at the Department of Automatic Control at Lund University. Without the inspiring discussions over coffee the quality would definitely have suffered.

Contents

1. Introduction 9

1.1 Background . . . 9
1.2 An introductory example . . . 10
1.3 Aim of thesis . . . 11
1.4 Outline . . . 12
1.5 Scientific contribution . . . 13

2. Background 14

2.1 Youla parametrization . . . 14
2.2 Generalized plant . . . 14
2.3 Convex optimization . . . 16
2.4 Linear algebra . . . 17
2.5 Inner products on $\mathcal{L}_2$ space . . . 20
2.6 System norms . . . 22
2.7 Basis functions . . . 23
2.8 LQG . . . 25
2.9 Constraints on the step response . . . 26
2.10 Constraints on the control signal . . . 27
2.11 Design criteria . . . 28

3. Analysis and computation of the cost functional 29

3.1 Hessian structure . . . 29
3.2 Inner products using state-space methods . . . 31
3.3 State-space based construction of the quadratic program . . . 33
3.4 State-space realization of the simplified Laguerre filters . . . 34
3.5 State-space realization of the Laguerre filters . . . 36
3.6 Output-orthogonal realization of the Laguerre filters . . . 38
3.7 Lyapunov equations exploiting Toeplitz structure . . . 41
3.8 Comparison — numerical example . . . 42

4. Extension to MIMO systems 45

4.1 Vectorizing the transfer Matrix . . . 45
4.2 Constructing the cost function . . . 47

5. Results **49**
5.1 MIGO test-batch 49
5.2 Effects of regularization 56
5.3 Flatbed test-case 57
6. Discussion **62**
6.1 Convergence 62
6.2 Numerical issues 62
6.3 Applications in other areas 62
6.4 Suggested further research 63
6.5 Conclusion 65
Bibliography **66**

1

Introduction

1.1 Background

Linear-controller design is a fundamental part of automatic control, both in terms of research and industrial applications. It is thus of great interest to find the best possible linear regulator. The unconstrained case with Gaussian disturbance and quadratic cost has an analytical finite-dimensional solution, but might not be a good representation of the constraints we would like to place on a system.

Through the works of [Youla et al., 1976a; Youla et al., 1976b; Boyd and Barratt, 1991] we know that the linear regulator problem has an affine representation and that several of the most common design criteria and cost functionals are convex, or have convex approximations. Convexity of cost and constraints implies that we can construct a convex optimization problem. Solving the convex optimization problem would then give us either the best possible linear controller, or the information that the problem has no feasible solution. In order to use the readily available solvers for convex optimization problems it is required to find finite-dimensional representations of the originally infinite-dimensional problem. [Megritski, 1994] showed that in general the mixed $\mathcal{H}^2/\mathcal{H}_\infty$ problem has a solution of infinite order, and that order constraints needs to be considered in junction to the original problem. [Sznaier, 1994; Wernrud, 2008] covers to some extend discrete-time problems, and [Sznaier, 2000] covers continuous systems by projecting onto Laguerre filters and transforming into discrete-time domain by use of bilinear transform. One of the main difficulties when performing Youla-Parametrized controller design are numerical issues, which is the topic of this thesis. The aim of this thesis is to propose numerically robust and computationally efficient algorithms to transform quadratic cost-functionals on the subspace of $\mathcal{H}_2$ spanned by a truncated Ritz expansion to equivalent cost functions on the finite-dimensional space of coefficient vectors.

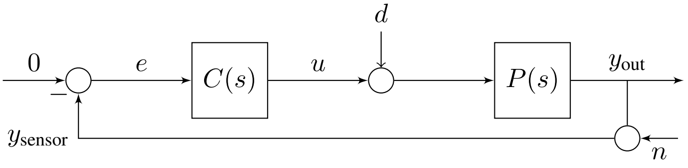


Figure 1.1 Block diagram of a 1-DOF system, with 0 reference input

1.2 An introductory example

The following example is intended to provide a quick overview of what is to be treated in this thesis. The main purpose is to illustrate that even simple, unconstrained problems might give rise to numerical issues, which motivates the aim of this thesis.

Example 1.1—Resonant system

The system considered is illustrated in Figure 1.1. The plant transfer function as well as the spectral factors of the measurement noise and load disturbance are given in (1.1).

$$\begin{aligned} P(s) &= \frac{1}{(s+1)(s^2+2\cdot 0.01s+1)} \\ D(s) &= \frac{1}{s/0.1+1} \\ N(s) &= 1 \end{aligned} \tag{1.1}$$

where $P(s)$ is the open-loop transfer function of the plant, $D(s)$ is the spectral factor of the process disturbance (such that $D(-s)D(s) = \Phi_d(s)$ where $\Phi_d(s)$ is the spectral density of the process disturbance), and $N(s)$ is a spectral factor of the measurement noise. The intensity of the process disturbance is $W_{\text{proc}}^2 = 20^2$ and the intensity of the measurement noise disturbance is $W_{\text{sensor}}^2 = (10^{-5})^2$.

Problem formulation

$$\underset{Q}{\text{minimize}}\; J_{\text{lqg}} = \int_0^\infty |y_{\text{out}}(t)|^2 + \rho|u(t)|^2 dt$$

where $\rho = 0.3$.

Analytical solution This problem is an LQG problem and has a closed form solution resulting in $J_{\text{lqg}}^* = 4.8$.

Through convex optimization In order to use convex optimization techniques we need to reduce the problem from infinite-dimensional, to a finite-dimensional problem, so that the problem is changed from finding a function, to finding a vector of coefficients. We do this by expanding $Q(s)$ in a complete basis, and then truncating to get

$$Q(s) = \sum_{i=1}^{N} q_i(s)\beta_i$$

We consider the following basis functions: $q_i(s) = \left(\frac{1}{s+a}\right)^i$ where a is a positive real constant. J_{lqg} can be rewritten in frequency domain as:

$$J_{\text{lqg}} = \frac{1}{2\pi}\int_{-\infty}^{\infty} |H_{y_{\text{out}}d}(i\omega)|^2 + |H_{y_{\text{out}}n}(i\omega)|^2 + \rho\left(|H_{ud}(i\omega)|^2 + |H_{un}(i\omega)|^2\right) d\omega \quad (1.2)$$

The resulting convex optimization problem becomes

$$\underset{\beta}{\text{minimize}} \quad J_{\text{lqg}} \qquad \square$$

The results of solving the problem[1] for different cutoff frequencies and 15 elements in the basis expansion is shown in Figure 1.2. We will focus on two issues visible in this figure: The spikes in the range $a \in (1, 5)$, this types of numerical issues will be resolved in Chapter 3; the solution does not converge to optimum for certain ranges of a. This will be discussed in Chapter 6.

1.3 Aim of thesis

Example 1.1 illustrates and motivates the questions that this thesis aims to answer.

1. Why does numerical issues occur when constructing the cost function?
2. Can we develop algorithms which are more robust?
3. How does the cutoff frequency of the elements in the Ritz expansion influence convergence to optimality?

One way to gain a better understanding of how the cutoff frequency influences the rate of convergence is to solve several problems using different cutoff frequencies and numbers of basis elements. In order for this to be a tractable approach we need to find computationally efficient algorithms which allows for a wide range of cutoff frequencies and a sufficient number of elements in the basis expansion.

[1] The frequency integral (1.2) was numerically computed by use of the trapezoidal rule, and 2000 logarithmically spaced points in $[10^{-8}, 10^{6}]$. Setting $N = 15$, sweeping for values of $a \in [0.1, 1000]$. The resulting convex optimization problems were solved by performing Cholesky decomposition on the Hessians, and solving equivalent second order cone programs. All of which will be covered in Chapter 2

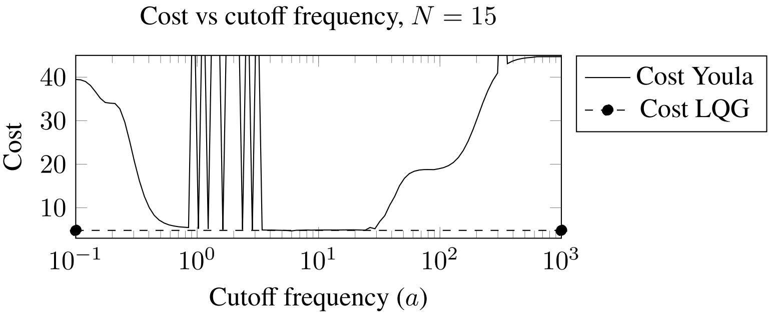


Figure 1.2 Cost plotted against the cutoff frequency (a), using 15 elements in the basis expansion. Note the spikes, and that convergence seems dependent on the cutoff frequency.

1.4 Outline

Chapter 2 covers the theory necessary to comfortably follow the reasoning in the following chapters. We start by covering Youla parametrization, convex optimization, linear algebra, and some interesting results of Pascal matrices which are used in Chapter 3. Inner products, system norms, basis functions, LQG and some convex constraints are briefly covered.

Chapter 3 concerns analysis and computation of finite-dimensional approximations of the cost functional, especially using Laguerre bases for the Ritz expansion. We start by analyzing the Hessian, which is proven to be Toeplitz if Laguerre bases are used. Thereafter we propose to compute inner products via solving Sylvester equations in Theorem 3.4. In Section 3.3 we demonstrate how to apply said theorem to construct quadratic-program approximations of $\mathcal{H}_2$ cost functionals. Sections 3.4–3.6 concerns the properties and effects of applying different state-space realizations, culminating in a closed form expression of the output-orthogonal realization. In Section 3.7 we construct an algorithm to exploit Toeplitz structure when solving Lyapunov equations.

Extension to MIMO systems are covered in Chapter 4 which also covers how to reduce the number of Lyapunov equations one needs to solve in order to construct the quadratic program.

In Chapter 5 we evaluate four algorithms based on the theory presented in Chapters 3 and 4 on some problems present in earlier research. The focus lies on recovering LQG controllers, and presenting exact solutions to finite-dimensional approximations of LQG type cost functionals for systems with time delay.

Finally present our conclusions and suggestions for further research in Chapter 6.

1.5 Scientific contribution

The scientific contribution can be divided into two parts, analysis and algorithm development. In Chapter 3 we show that the simplified Laguerre bases leads to ill-conditioned cost functionals, that a Jordan realization of Laguerre Filters will have ill-conditioned B matrices, and that the output-orthogonal realization of Laguerre Filters exhibits satisfactory matrix properties. Furthermore we derive how to calculate inner products on $\mathcal{H}_2$ using state-space methods, which is used in Chapter 4 to derive a quick and efficient algorithm to construct a quadratic program equivalent to a quadratic cost-functional on $\mathcal{H}_2$ for MIMO systems.

2

Background

2.1 Youla parametrization

This thesis is based on what in control literature is known as Youla parametrization, or Q-parametrization. Youla parametrization is a method to parametrize all closed-loop stable transfer functions for an LTI system obtainable by a LTI controller. Let C be the controller and Q be the Youla parameter. The set of all rational stabilizing controllers for the stable plant $P(s)$ can then be defined as

$$\left\{ \frac{Q(s)}{1 - P(s)Q(s)}, \quad Q(s) \in \Omega \right\}$$

where Ω is the set of proper stable rational transfer functions.

Gang of four

Using Youla parameterization all internal closed-loop transfer functions for a stable system (gang of four) becomes affine in $Q(s)$. Consider again the system depicted in Figure 1.1. The transfer function from process disturbance to output is given by $H_{y_{\text{out}}d}(s) = P(s)(1 - P(s)Q(s))$, from measurement noise to output $H_{y_{\text{out}}n}(s) = P(s)Q(s)$, from process disturbance to actuator signal $H_{ud} = P(s)Q(s)$, from measurement noise to control signal $H_{un} = Q(s)$, and the sensitivity function is given by $S(s) = 1 - P(s)Q(s)$.

2.2 Generalized plant

This subsection briefly covers the input-output feedback model known as the generalized plant. Collect all exogenous inputs into w, and the actuator signals in u, the regularized output in z and the measured signals in y. The structure is depicted in Figure 2.1 where

$$\tilde{P}(s) = \begin{bmatrix} P_{zw}(s) & P_{zu}(s) \\ P_{yw}(s) & P_{yu}(s) \end{bmatrix}$$

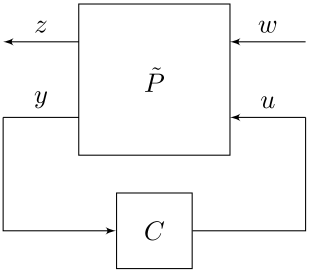


Figure 2.1 Generalized plant

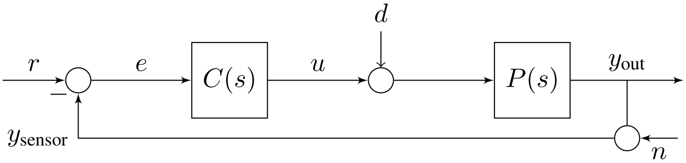


Figure 2.2 Standard 1-DOF system.

where P_{ab} is the open loop transfer function from b to a and assumed stable. The transfer function matrix of the closed-loop system is [Boyd and Barratt, 1991]

$$H_{zw}(s) = P_{zw}(s) - P_{zu}C(I - P_{yu}(s)C)^{-1}P_{yw}$$

which using Youla parametrization becomes

$$H_{zw}(s) = P_{zw}(s) - P_{zu}Q(s)P_{yw}$$

where $Q(s) = C(I - P_{yu}(s)C)^{-1}$. The next example is intended to clarify the concept

Example 2.1—Generalized plant

Consider the feedback system shown in Figure 2.2. Transforming to the general plant model can be done by collecting the exogenous inputs and placing them in $w = [d\ n\ r]^T$, collecting the regulated outputs and placing them in $z = [y_{\text{out}}\ u]^T$, and collecting the measurement signals and placing them in $e = r - y_{\text{sensor}} - n$. Thus

$$\tilde{P}(s) = \left[\begin{array}{c|c} P_{zw} & P_{zu} \\ \hline P_{yw} & P_{yu} \end{array}\right] = \left[\begin{array}{ccc|c} P & 0 & 0 & P \\ 0 & 0 & 0 & I \\ \hline -P & -I & I & -P \end{array}\right]$$ □

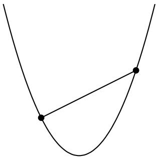


Figure 2.3 The graph of a convex function $\big(f(x) = x^2\big)$. Note that a line segment between two points of the graph lies above the graph of the function.

Note that all the interal closed-loop transfer functions, except of the sensitivity function are present in H_{zw}. The post multiplication of a transfer matrix is somewhat inconvenient, and it is possible to show that all internal transfer functions can be written on the form [Wernrud, 2008; Sznaier, 2000]

$$S(s) = S_0(s) + S_1(s)Q(s)$$

A slightly different approach to extend into the realm of MIMO systems than that of [Wernrud, 2008; Sznaier, 2000], which is more closely tied to the methods discussed in this thesis is presented in Chapter 4.

2.3 Convex optimization

Convex optimization has been, and continues to be a hot topic of research. The main reason is that convexity guarantees global solutions of minimization problems. This section is intended to provide the reader with a quick overview of two standard problem formulations frequently encountered in convex optimization, which are used heavily in the later chapters of this thesis. The following definition is taken from [Boyd and Vandenberge, 2009].

Definition 2.1—Convex function
A function $f : \mathbf{R}^n \to \mathbf{R}$ is convex if dom$f$ is a convex set and if $\forall x,\ y \in$ dom$f$, and $\theta \in [0,\ 1]$

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y) \qquad \square$$

In essence, this means that the line segment between two points of the function should lie above the function itself, see Figure 2.3

QP - quadratic program

One of the standard problems, which are readily solved by most solvers is the quadratic program

DEFINITION 2.2—QP

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array} \tag{2.1}$$

where $P_i,\ i = 0,\ 1\ \ldots,\ m$ are positive semidefinite matrices. □

SOCP - second order cone program

A more general problem than the quadratic program is the second order cone program.

DEFINITION 2.3—SOCP

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m \\ & Fx = g \end{array} \tag{2.2}$$

□

Solvers

There exists several solvers for QP and SOCP on the market. Throughout this thesis we use [MOSEK, 2018]. Some alternative solvers are [Gurobi, 2018], which can solve quadratically constrained quadratic programs (QP), [CPLEX, 2018] which solves both QP and SOCP, and the [MathWorks, 2018]. Directly calling these solvers can be complicated. A convenient alternative is to use an interface such as [CVX, 2018] which simplifies the process. However, in order to track and analyze every step of the process we have called MOSEK directly.

2.4 Linear algebra

This subsection is intended to familiarize the reader with some of the definitions, and common results frequently used in this thesis.

DEFINITION 2.4—POSITIVE (SEMI)DEFINITENESS
A matrix A is positive (semi)definite if and only if

- $x^T A x > (\geq) 0, \qquad \forall x \neq 0$
- $\lambda_i(A) > (\geq) 0, \qquad \forall i$, where λ_i denotes the i:th eigenvalue of A □

Definition 2.5—Frobenius Norm
The Frobenius norm, denoted $\|\cdot\|_{\text{fro}}$ is given by

$$\|A\|_{\text{fro}} = \sqrt{\operatorname{tr} A^T A} = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} a_{i,j}}$$

where tr is the trace operator, and $a_{i,j}$ is the element on the ith row and jth column of A. □

Definition 2.6—Toeplitz
A Toeplitz matrix (T) of dimension $n \times n$ is a square matrix which can be written on the form

$$T = \begin{bmatrix} t_0 & t_1 & \cdots & t_n \\ t_{-1} & t_0 & \ddots & t_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ t_{-n} & t_{-(n-1)} & \cdots & t_0 \end{bmatrix}$$ □

A symmetric Toeplitz matrix, is a Toeplitz matrix where $t_{-k} = t_k$.

Definition 2.7—Condition number
The condition number of a matrix A, denoted $\rho(\cdot)$ is defined as

$$\rho(A) = \frac{\max_i(\sigma_i(A))}{\min_i(\sigma_i(A)}$$

where $\sigma_i(A)$ is the i:th singular value of A. □

To see the relevance of condition numbers consider a linear system on the form

$$Ax = b$$

and let δ be the error in b, i.e. $b = b_0 + \delta$. Then an upper bound on the norm of the relative error becomes

$$\begin{aligned} \|e\|_2 &\leq max_{\delta,b\neq 0}\left(\frac{\|A^{-1}\delta\|_2}{\|\delta\|_2}\right) / \left(\frac{\|A^{-1}b\|_2}{\|b\|_2}\right) \\ &= \rho(A) \end{aligned}$$

Pascal matrices

It will be helpful to briefly discuss the properties of Pascal matrices, as they occur frequently in this thesis. The definitions, theorems and corresponding proofs can be found in [Brawer and Pirovino, 1992]. We will start by defining the triangular Pascal matrix and the symmetric Pascal matrix, followed by a quick summary of the properties most relevant to this thesis.

Definition 2.8—Triangular Pascal matrix

The $n \times n$ lower triangular Pascal matrix P_L is defined by

$$P_L(i,j) = \begin{cases} \binom{i}{j}, & i,j = 0,\ldots,n-1, \quad j < i \\ 0, & j > i \end{cases} \qquad \square$$

Example 2.2—P_L

The lower-triangular Pascal matrix of dimension 4×4 becomes

$$P_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 1 & 3 & 3 & 1 \end{bmatrix} \qquad \square$$

Note that the lower-triangular Pascal matrix has binomial expansion coefficients as rows.

Definition 2.9—Symmetric Pascal matrix

The symmetric Pascal matrix P_S of dimension $n \times n$ is defined by

$$P_S(i,j) = \begin{pmatrix} i+j \\ j \end{pmatrix}, \; i,j = 0,\ldots,n-1 \qquad \square$$

Example 2.3—P_S

The symmetric Pascal matrix P_S of dimension 4×4 is

$$P_S = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix} \qquad \square$$

Note that the symmetric Pascal matrix has binomial expansion coefficients as anti-diagonals.

Properties The following theorem will be used in Chapter 3 to analyze the numerical properties of state-space realizations of Laguerre filters.

Theorem 2.1—Inverse of lower triangular Pascal matrix

The inverse of P_L is

$$D_{-1} P_L D_{-1}$$

where $D_{-1} = \mathrm{diag}[1 \;\; -1 \;\; \ldots \;\; (-1)^{n-1}]$

Proof See [Brawer and Pirovino, 1992]. □

The following theorem is not necesseary to understand this thesis but given for completeness sake.

THEOREM 2.2—CHOLESKY FACTORIZATION OF P_S
The Cholesky factorization of P_S is given by

$$P_S = P_L P_L^T \qquad \square$$

Proof See [Brawer and Pirovino, 1992]. □

2.5 Inner products on $\mathcal{L}_2$ space

Throughout this section lower case letters will be used for functions in time domain whereas upper case letters will be used for functions in frequency domain.

Time domain

Let $\mathcal{L}_2(\mathbb{R}^+)$ denote the Hilbert space of square integrable functions on $\mathbb{R}^+$, equipped with the inner product

$$\langle f, g\rangle = \int_0^\infty f(t)^* g(t) dt$$

Frequency domain

Let $\mathcal{L}_2(i\mathbb{R})$ denote the Hilbert space of square integrable functions on $i\mathbb{R}$, equipped with the inner product

$$\langle F, G\rangle = \frac{1}{2\pi}\int_{-\infty}^\infty F^*(i\omega)G(i\omega)d\omega$$

If F and G are analytic, and vanish at infinity this can be calculated using calculus of residues

$$\begin{aligned}\langle F, G\rangle &= \frac{1}{2\pi}\int_{-\infty}^\infty F^*(i\omega)G(i\omega)d\omega \\ &= \frac{1}{2\pi i}\oint_{\mathcal{C}} F^*(-s^*)G(s)ds \\ &= \sum_{i=1}^{k} \operatorname{Res}\left[F^*(-s^*)G(s); p_i\right]\end{aligned} \qquad (2.3)$$

where $\{p_i\}_{i=1}^k$ is the set of poles encircled by $\mathcal{C}$, and $\mathcal{C}$ is chosen such that the entire imaginary axis is included, see Figure 2.4. Since $F,\ G$ vanish at infinity the contribution there is zero.

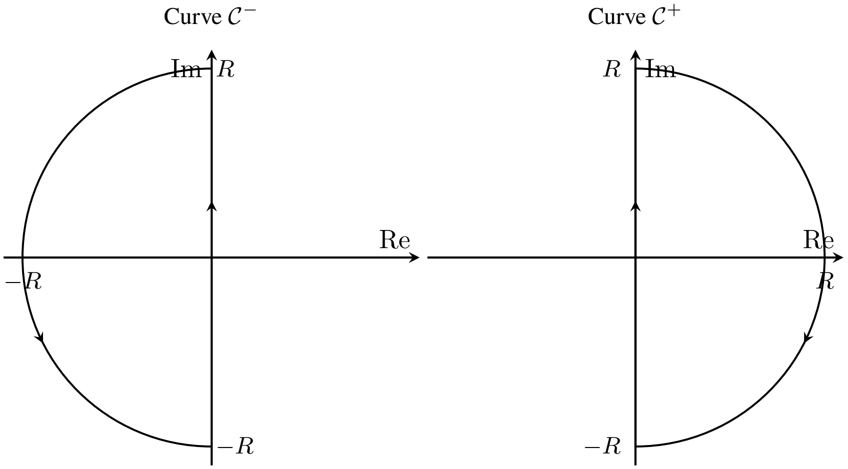


Figure 2.4 Illustration of the extension of integrating over the imaginary line to a curve integral, where $R \to \infty$.

Calculating the residues

Through partial fraction expansion Since the residue at a pole is the coefficient associated with the first term in the Laurent series expansion around that pole we can do a partial fraction expansion of $F^*(-s^*)G(s)$. Consider

$$F^*(-s^*)G(s) = \frac{c_1^{(1)}}{s+p_1} + \frac{c_1^{(2)}}{(s+p_1)^2} + \cdots + \frac{c_1^{(n_1)}}{(s+p_1)^{n_1}} + \frac{c_2^{(1)}}{1+p_2} + \cdots$$

Then $\langle F, G\rangle = \sum(c_i^{(1)})$

Using the Limit theorem The residue at a simple pole p the residue of F is given by

$$\text{Res}[F;p] = \lim_{s\to p}(s-p)F(s)$$

For higher order poles we can use the corresponding limit formula. Let p be a pole of order n, then the residue around p can be calculated from

$$\text{Res}[F;p] = \frac{1}{(n-1)!}\lim_{s\to p}\frac{\mathrm{d}^{n-1}}{\mathrm{d}s^{n-1}}\left((s-p)^n F(s)\right) \tag{2.4}$$

Differentiating rational functions Let

$$F(s) = \frac{\prod_{i=1}^{N_z}(s-z_i)}{\prod_{i=1}^{N_p}(s-p_i)}$$

where N_z and N_p are the number of zeros and poles respectively. Poles and zeros of higher multiplicity are treated as multiplication of separate poles, i.e. $(s-p)^2 = (s-p_1)(s-p_2)$ with $p_1 = p_2 = p$. Differentiating once with respect to s gives [Ma et al., 2014]

$$
\begin{aligned}
F^{(1)}(s) &= \frac{\begin{gathered}\sum_{i=1}^{N_z}\left(\prod_{j\neq i}(s-z_j)\right)\prod_{k=1}^{N_p}(s-p_k)\\ -\prod_{i=1}^{N_s}(s-z_i)\sum_{j=1}^{N_p}\left(\prod_{k\neq j}(s-p_k)\right)\end{gathered}}{\left(\prod_{i=1}^{N_p}(s-p_i)\right)^2}\\
&= \frac{\sum_{i=1}^{N_z}\left((s-z_i)^{-1}\right)\prod_{i=1}^{N_i}(s-z_i)-\sum_{i=1}^{N_p}\left((s-p_i)^{-1}\right)\prod_{i=1}^{N_z}(s-z_i)}{\prod_{i=1}^{N_p}(s-p_i)}\\
&= \underbrace{\frac{\prod_{i=1}^{N_z}(s-z_i)}{\prod_{i=1}^{N_p}(s-p_i)}}_{F(s)}\underbrace{\left(\sum_{i=1}^{N_z}\frac{1}{s-z_i}-\sum_{i=1}^{N_p}\frac{1}{s-p_i}\right)}_{R(s)}
\end{aligned}
$$

Repeatedly applying the product rule gives

$$F^{(k)}(s) = \sum_{j=1}^{k-1}\binom{k}{j}F^{(k-j)}(s)R^{(j)}(s) \tag{2.5}$$

where

$$R^{(k)}(s) = (-1)^k k!\left(\sum_{i=1}^{N_z}\frac{1}{(s+z_i)^{k+1}}-\sum_{j=1}^{N_p}\frac{1}{(s+p_j)^{k+1}}\right)$$

and $F^(k)(s)$ denotes the kth derivative of F with respect to s.

2.6 System norms

$\mathcal{H}_2$ norm

Let a stable SISO system

$$
\begin{aligned}
\dot{x} &= Ax + B\\
y &= Cx
\end{aligned}
$$

have transfer function $H(s)$ and impulse response $h(t) = Ce^{At}B$. The $\mathcal{H}_2$ norm is then given by

$$\|H\|_2 = \left(\frac{1}{2\pi}\int_{-\infty}^{\infty}|H(i\omega)|^2 d\omega\right)^{1/2}$$

which is conveniently written in terms of inner products as $\|H\|_2 = \sqrt{\langle H, H\rangle} = \sqrt{\langle h, h\rangle}$. The norm can be calculated through contour integration and calculus of residues, but if H is a stable system the preferred method of calculation is by using the impulse response in the following manner

$$\begin{aligned}\langle h, h\rangle &= \int_0^\infty \left(Ce^{At}B\right)^T \left(Ce^{At}B\right) dt \\ &= B^T \underbrace{\int_0^\infty e^{A^T t} C^T C e^{At} dt}_{\mathbf{W}_o} B\end{aligned}$$

Assuming A to be stable, it is easy to verify that

$$A^T\mathbf{W}_o + \mathbf{W}_o A + C^T C = 0$$

where $\mathbf{W}_o$ is the Observability Gramian, thus $\|H\|_2 = \sqrt{B^T\mathbf{W}_o B}$. The $\mathcal{H}_2$ norm can be interpreted as the energy of the impulse response.

$\mathcal{H}_\infty$ norm

For a linear system $H(s)$ we have [Zhou and Doyle, 1997]

$$\|H\|_\infty = \sup_{Re(s)>0} \bar{\sigma}[H(s)] = \sup_{\omega \in R} \bar{\sigma}[H(i\omega)]$$

where $\bar{\sigma}$ is the maximum singular value. The $\mathcal{H}_\infty$ norm is best understood as the maximum possible amplification of signals. For SISO systems it translates to the largest possible amplification over all frequencies, for MIMO systems we need to consider the input configurations which gives the largest possible amplifications, which motivates the usage of the largest singular value.

2.7 Basis functions

Definition 2.10—Laguerre filter
A Laguerre filter of order k is given by

$$q_k(s) = \frac{\sqrt{2a}}{a+s}\left(\frac{a-s}{a+s}\right)^k \qquad \square$$

where $a > 0$.

Theorem 2.3—Completeness
The Laguerre filters constitute a complete basis

Proof see [Heuberger et al., 2005]. □

LEMMA 2.4—ORTHONORMALITY
The Laguerre filters defined in Definition 2.10 constitute an orthonormal set of functions in Laplace domain.

Proof

$$q_j(-s)q_k(s) = 2a\frac{(a-s)^{k-j-1}}{(a+s)^{k-j+1}} \tag{2.6}$$

$$\begin{aligned}\langle q_j, q_k\rangle &= \frac{1}{2\pi i}\oint q_j(-s)q_k(s)ds \\ &= \sum_{i=1}^{m} \text{Res}[q_j(-s)q_k(s); p_i]\end{aligned}$$

where $\{p_i\}_{i=1}^m$ is the set of poles encircled by the contour, and the contour is taken around either complex half plane. By inspecting (2.6) it is clear that for $j < k$ the integrand has no poles i the right half plane, whether as for $j > k$ the integrand has no poles in the left half plane. Thus the integral must be zero, the only remaining non-zero possibility is $j = k$ which results in

$$\begin{aligned}\langle q_j, q_j\rangle &= \text{Res}\left[\frac{2a}{(a-s)(a+s)}\right] \\ &= 2a \lim_{s\to a} \frac{a-s}{(a-s)(a+s)} \quad = 1\end{aligned}$$

which concludes the proof. □

Simplified Laguerre basis

Removing all the zeros from the Laguerre filters leaves us with a series of poles. These filters will be defined as the simplified Laguerre filters, and were mentioned as a candidate for Youla-parameterization based methods in [Boyd and Barratt, 1991].

DEFINITION 2.11—SIMPLIFIED LAGUERRE FILTERS
A simplified Laguerre filter of order k is given by

$$q_k(s) = \frac{1}{(s+a)^k}$$ □

where $a > 0$.

Even though the simplified Laguerre filters are not orthonormal, they constitute a complete basis. A sequence of the N first simplified Laguerre filters span the same space as the sequence of the N first Laguerre filters. This can be seen by performing a partial fraction expansion of the Laguerre filter, which becomes a projection onto the simplified Laguerre filters.

2.8 LQG

In this section we will assume that the system, along with load disturbance d and sensor noise n has a realization on the form

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_d v_1 \\ y_{\text{out}} &= C_{\text{out}} x \\ y_{\text{sensor}} &= C_{\text{sensor}} x + v_2 \end{aligned} \tag{2.7}$$

where $v_1,\ v_2$ are white noise, y_{out} is the regulated output and y_{sensor} are the sensor signals. The importance of the following theorem lies in that it is the optimal linear controller for the unconstrained case, thus we are going to evaluate our algorithms as to whether we are able to recover the LQG controller, or not. LQG design also works as a simple method to stabilize an unstable plant so that we may use Youla-parameter design methods on all stabilizable and detectable systems.

THEOREM 2.5—LQG
Consider the system (2.7) where $\begin{bmatrix} v_1^T & v_2^T \end{bmatrix}^T$ is white noise with intensity

$$R = \begin{bmatrix} R_1 & R_{12} \\ R_{12}^T & R_2 \end{bmatrix}$$

The goal is to find the linear feedback law $u(s) = -F_y(s) y_{\text{sensor}}(s)$ which minimizes

$$J_{\text{lqg}} = \int_0^\infty z^T(t) Q_1(t) z(t) + 2x^T(t) Q_{12} u(t) + u^T(t) Q_2 u(t) dt$$

where Q_2 is positive definite,

$$\begin{bmatrix} C_{\text{out}}^T Q_1 C_{\text{out}} & Q_{12} \\ Q_{12}^T & Q_2 \end{bmatrix}$$

is positive semidefinite, (A, B_u) is a stabilizable pair, and $(A, C_{\text{out}}^T Q_1 C_{\text{out}})$ is a detectable pair. R_2 is positive definite and R is positive semidefinite. Assume (A, C_{sensor}) are detectable and $(A - R_{12} R_2^{-1} C_{\text{sensor}}, R_1 - R_{12} R_2^{-1} R_{12}^T)$ are stabilizable. The optimal linear controller is then given by

$$\begin{aligned} u(t) &= -L\hat{x}(t) \\ \hat{x}(t) &= A\hat{x}(t) + B_u u(t) + K(y(t) - C_{\text{sensor}} \hat{x}(t) \end{aligned}$$

where L can be obtained using `lqr` and K using `kalman` commands in `MATLAB`. For proofs and explicit expressions of L and K, see [Glad and Ljung, 2016]. □

Norm representation

The purpose of this subsection is to explain how to cast an LQG-criteria as a convex optimization problem. We sacrifice generality to focus on the criteria mostly used in this thesis. Let $z = [u\ y_{\text{out}}]^T$, where y_{out} is the regulated output. The cost considered is

$$J_{\text{lqg}} = \int_0^\infty |y_{\text{out}}(t)|^2 + \rho |u(t)|^2 dt$$

Taking

$$V = \begin{bmatrix} 1 & 0 \\ 0 & \rho \end{bmatrix}$$

lets us write J_{lqg} as

$$\begin{aligned} J_{\text{lqg}} &= \text{tr}\left\{ V \int_0^\infty z(t) z^*(t) \right\} \\ &= \frac{1}{2\pi} \text{tr}\left\{ V \int_{-\infty}^\infty Z(i\omega) Z^*(i\omega) d\omega \right\} \\ &= \frac{1}{2\pi} \text{tr}\left\{ V \int_{-\infty}^\infty H_{zw}(i\omega) \Phi_w H^*(i\omega) \right\} \end{aligned}$$

where $Z(i\omega)$ denotes the frequency domain transform of $z(t)$, Φ_w denotes the power spectral density of $w = [d\ n]^T$ where d is the process disturbance and n is the measurement noise (seen as a stationary stochastic processes), and $H_{zw}(i\omega)$ is the transfer function matrix from w to z. Assuming uncorrelated noises, factor Φ_w in the following manner

$$\Phi_w(i\omega) = \begin{bmatrix} D(i\omega) D^*(i\omega) & 0 \\ 0 & N(i\omega) N^*(i\omega) \end{bmatrix}$$

where $D(i\omega)$ and $N(i\omega)$ are spectral factors of the process disturbance spectral density and the measurement noise spectral density respectively. Expanding the trace of the matrix results in

$$J_{\text{lqg}} = \|H_{y_{\text{out}}d} D\|_2^2 + \|H_{y_{\text{out}}n} N\|_2^2 + \rho \left(\|H_{ud} D\|_2^2 + \|H_{un} N\|_2^2 \right) \tag{2.8}$$

2.9 Constraints on the step response

Rise time, settling time, overshoot and undershoot can all be addressed by creating time dependent upper and lower limits for the step response (or any other response for that matter). The idea is to view the system as a multi input single output system

$$T(s) r = P(s) [q_1(s)\ q_2(s)\ \ldots\ q_N(s)] (\beta r)$$

the system is then realizable on the form

$$\begin{aligned}\dot{x} &= Ax + B\beta r\\ z &= Cx\end{aligned}$$

with the response to a given input $r(t)$

$$z(t) = C\int_0^t r(t)e^{A(t-\tau)}d\tau B\beta$$

and constraints on the form

$$z_{\text{min}}(t) \leq z(t) \leq z_{\text{max}}(t)$$

Rise time

Let z be the controlled output, and $r(t)$ be a step function. Also define the rise time as the time from $t = 0$ to $z(t) = \gamma_{\text{rt}}$. Setting a maximum rise time to t_{rt} could be specified as $z_{\text{min}}(t) = \gamma_{\text{rt}}$ for $t \geq t_{rt}$.

Overshoot

Letting r be the unity step input, requiring the maximum overshoot to be less the γ_{OS} can be translated to choosing $z_{\text{max}} = \gamma_{\text{OS}},\ \forall t$.

Undershoot

Once again let r be the unity step input. A maximum undershoot of γ_{US} corresponds to setting $z_{\text{min}}(t) = -\gamma_{\text{US}},\ \forall t$.

Settling time

A $\gamma_{\text{settle}}\%$ settling time of less than t_{settle} can be constructed from

$$\left.\begin{aligned}z_{\text{min}}(t) &= 1 - \gamma_{\text{settle}}\\ z_{\text{max}}(t) &= 1 + \gamma_{\text{settle}}\end{aligned}\right\}\quad \forall\, t \geq t_{\text{settle}}$$

2.10 Constraints on the control signal

It is fairly straight forward to calculate the actuator response to different types of input, using the following relations

- Process disturbance to control signal: $H_{ud}(s) = P(s)Q(s)$

- Measurement noise and reference signal to control signal: $H_{un}(s) = Q(s)$

The key idea is to do a state-space realization on a form

$$\begin{aligned} \dot{x} &= Ax + B\beta r \\ u &= Cx \end{aligned}$$

This would allow you to simulate for a given input or disturbance

$$u(t) = \left(C \int_0^t r(t) e^{A(t-\tau)} d\tau B \right) \beta$$

2.11 Design criteria

When working with convex optimization we are presented with a common framework to deal with several cost criteria and constraints. In this thesis the main focus is in recovering the LQG controller presented in Section 2.8. The reason for this is that it is a fairly simple problem, with a true optimal solution, and that numerical issues are apparent. Before exploring more exotic cost functionals, it would make sense to find methods which allows us to consistently handle simple cost functionals. Design specifications which can be handled by use of Youla parametrization and convex optimization can be represented as a convex optimization problem [Boyd and Barratt, 1991]

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \underbrace{J(\beta)}_{\text{convex}} \\ \text{subject to} \quad & \underbrace{G_i(\beta)}_{\text{convex}} \leq 0 \\ & A\beta + b = 0 \end{aligned}$$

where A is a matrix and b is a vector. In this paper we mainly focus on the unconstrained case.

Cost functionals

Other than $\mathcal{H}_2$ minimization, other common cost functionals are integrated absolute error, $\mathcal{H}_\infty$ norm of inner closed-loop transfer functions.

3

Analysis and computation of the cost functional

This chapter covers the analysis of cost functions on the finite-dimensional space of coefficient vectors, equivalent to cost functionals on the subspace of $\mathcal{H}_2$ spanned by a truncated Ritz expansion. Consider the SISO 1-DOF case in Figure 1.1. The interesting transfer functions for a stable system are affine in the Youla parameter Q as introduced in Chapter 2. The purpose of this chapter is to introduce algorithms that reduce (3.1) to (3.2)

$$J = a\|H_{y_{\text{out}}d}\|_2^2 + b\|H_{y_{\text{out}}n}\|_2^2 + c\|H_{ud}\|_2^2 + d\|H_{un}\|_2^2 \tag{3.1}$$

$$J = \beta^T M \beta + q_0^T \beta + r_0 \tag{3.2}$$

The main result is the state-space based construction of the quadratic program (3.2) described in Section 3.3, which builds on Theorem 3.4 that states that inner products between two stable systems can be calculated via solving a Sylvester equation. The state-space based construction is then applied to show that the Simplified Laguerre filters leads to ill-conditioned cost functions, and that a potentially good approach is to use the Output Orthogonal Realization of Laguerre filters.

3.1 Hessian structure

By parameterizing Q using a Ritz expansion as in (3.3), the infinite-dimensional problem of selecting Q is reduced to the finite-dimensional problem of selecting the coefficient vector β

$$Q(s) = \sum_{i=1}^{N} \beta_i q_i(s) = q^T \beta \tag{3.3}$$

Dealing with the Ritz approximation $H_{y_{\text{out}}n}$, H_{ud} and H_{un} can be handled in similar ways, whereas the cross terms in $H_{y_{\text{out}}d}$ requires some extra attention. Consider

$$\begin{aligned}\|GQ\|_2^2 &= \langle Gq^T\beta, Gq^T\beta\rangle \\ &= \beta^T \underbrace{\begin{bmatrix} \langle Gq_1, Gq_1\rangle & \langle Gq_1, Gq_2\rangle & \cdots & \langle Gq_1, Gq_N\rangle \\ \langle Gq_2, Gq_1\rangle & \langle Gq_2, Gq_2\rangle & & \\ \vdots & & \ddots & \vdots \\ \langle Gq_N, Gq_1\rangle & \langle Gq_N, Gq_2\rangle & \cdots & \langle Gq_N, Gq_N\rangle \end{bmatrix}}_{M} \beta\end{aligned}$$

Using calculus of residues the elements of M can be calculated from

$$M_{j,k} = \langle Gq_j, Gq_k\rangle = \sum_i \operatorname{Res}\big[q_j^*(-s^*)G^*(-s^*)G(s)q_k(s); p_i\big]$$

where the sum is taken over all poles $\{p_i\}$ encircled by either $\mathcal{C}^-$ or $\mathcal{C}^+$.

Dealing with $H_{y_{out}d}$ Remembering $H_{y_{\text{out}}d}(s) = P(s)(1 - P(s)Q(s))$ we get

$$\begin{aligned}\|H_{y_{\text{out}}d}\|_2^2 &= \langle P(1-PQ), P(1-PQ)\rangle \\ &= \langle P, P\rangle + \underbrace{-\left(\langle P, PPQ\rangle + \langle PPQ, P\rangle\right)}_{J_{\text{lin}}} + \langle PPQ, PPQ\rangle\end{aligned}$$

the part linear in Q can then be expressed as $J_{\text{lin}} = q_0\beta$ with

$$\begin{aligned}q_0^{(j)} = -\Bigg(&\sum_i \operatorname{Res}[P^*(-s^*)P(s)q_j(s); p_i] \\ &+ \sum_i \operatorname{Res}\big[q_j^*(-s^*)P^*(-s^*)P(s); p_i\big]\Bigg)\end{aligned} \tag{3.4}$$

where $\|PPQ\|_2^2$ can be put on the form $\beta^T M\beta$ using the method described earlier. To handle J_{lin} note that the two terms in the integral give equal contributions since taking the contour integral over $\mathcal{C}^+$ is equivalent to taking the contour over $\mathcal{C}^-$ replacing s by $-s$.

Lemma 3.1—Hermitian
M is Hermitian ($M^* = M$).

Proof

$$M_{j,k} = \langle Gq_j, Gq_k\rangle = \langle Gq_k, Gq_j\rangle^* = M_{k,j}^* \qquad \square$$

LEMMA 3.2—POSITIVE SEMIDEFINITE
M is positive semidefinite.

Proof From the definition of norms we have that $\|GQ\|_2 \geq 0$. Since $\|QG\|_2^2 = \beta^T M \beta$, M is positive semidefinite. □

The Laguerre filters, see Definition 2.10 has an attractive property

LEMMA 3.3—M IS TOEPLITZ
For the Ritz expansion where

$$q_i(s) = \frac{\sqrt{2a}}{a+s}\left(\frac{a-s}{a+s}\right)^{i-1}$$

M is Toeplitz.

Proof The only part of $M_{j,k}$ which is dependent on the indices is the cross product between the basis elements which for the Laguerre filters becomes

$$q_j^*(-s^*)q_k(s) = 2a\frac{(a-s)^{k-j-1}}{(a+s)^{k-j+1}}$$

Note that this is only dependent on $k-j$, thus M is Toeplitz. □

3.2 Inner products using state-space methods

The idea to calculate $\|H\|_2$ using state-space methods is well accepted; the purpose of this section is to introduce state-space methods of calculating the inner product of any two stable, strictly proper and realizable transfer functions. This method handles input and output delays with little extra effort. Let H_1 and H_2 be transfer functions with impulse responses $h_1(t),\ h_2(t)$ with realizations

$$\mathcal{S}_i : \quad \begin{aligned} \dot{x}(t) &= A_i x(t) + B_i u(t) \\ y(t) &= C_i x(t-\tau_i) \end{aligned} \tag{3.5}$$

where A_i are asymptotically stable matrices and τ_i is the time delay of the system (here presented as output delay, but could also be input delay). Let $\hat{h}_i(t)$ be the impulse response of the undelayed system, i.e. $\hat{h}_i(t-\tau_i) = h_i(t)$, assume without loss of generality that $\tau_2 \geq \tau_1$. The scalar product between H_1 and H_2 is then given by

$$
\begin{aligned}
\langle H_1, H_2 \rangle &= \langle h_1, h_2 \rangle \\
&= \int_0^\infty h_1^*(t) h_2(t) dt \\
&= \int_0^\infty \hat{h}_1^*(t - \tau_1) \hat{h}_2(t - \tau_2) dt \\
&= \int_{\tau_2}^\infty \hat{h}_1^*(t - \tau_1) \hat{h}_2(t - \tau_2) dt \\
&= \int_0^\infty B_1^T e^{(t + \tau_2 - \tau_1) A_1^T} C_1^T C e^{A_2 t} B_2 dt
\end{aligned}
\tag{3.6}
$$

A word of caution is required here. Generally $e^X e^Y = e^{X+Y}$ does not hold in the case that X and Y are matrices. However if X, Y commutes the equality holds [Wahlén, 2014]. Since a matrix always commutes with itself we can factor the time invariant part outside of the integration.

$$
\langle H_1, H_2 \rangle = B_1^T e^{(\tau_2 - \tau_1) A_1^T} \underbrace{\int_0^\infty e^{tA_1^T} C_1^T C_2 e^{tA_2} dt}_{M_{1,2}} B_2
$$

Applying integration by parts results in

$$
\begin{aligned}
\int_0^\infty \frac{\mathrm{d}}{\mathrm{d}t} \left(e^{tA_1^T} C_1^T C_2 e^{tA_2} \right) dt = A_1^T \int_0^\infty & e^{tA_1} C_1^T C_2 e^{tA_2} dt \\
+ \int_0^\infty & e^{tA_1} C_1^T C_2 e^{tA_2} dt A_2 \\
-C_1^T C_2 = A_1^T M_{1,2} + M_{1,2} A_2 &
\end{aligned}
$$

This is summarized in Theorem 3.4.

Theorem 3.4—Inner product using State Space methods
Given two transfer functions H_1 and H_2, realizable on the form given in (3.5) the inner product can be calculated as

$$
\langle H_1, H_2 \rangle = \begin{cases} B_1^T e^{(\tau_2 - \tau_1) A_1^T} M_{1,2} B_2, & \tau_2 \geq \tau_1 \\ B_1^T M_{1,2} e^{(\tau_1 - \tau_2) A_2} B_2, & \tau_1 \geq \tau_2 \end{cases}
$$

where $M_{1,2}$ is the solution to the Sylvester equation

$$
A_1^T M_{1,2} + M_{1,2} A_2 + C_1^T C_2 = 0 \qquad \square
$$

note that if $H_1 = H_2$ Theorem 3.4 reduces to solving for the Observability Gramian of $\mathcal{S}_1$ and pre/post multiplying with B_1, which gives the standard result for calculating the $\mathcal{H}_2$ norm.

3.3 State-space based construction of the quadratic program

The purpose of this section is to introduce a method to construct a quadratic program by using Theorem 3.4. Consider $\|GQ\|_2$, where Q is parametrized by a Ritz expansion realized on the form

$$\begin{aligned}\dot{x}_Q(t) &= A_Q x_Q(t) + B_Q \beta r(t)\\ u(t) &= C_Q x_Q(t)\end{aligned} \tag{3.7}$$

and that the system is stable, with the realization

$$\begin{aligned}\dot{x}_G(t) &= A_G x_G(t) + B_G u(t)\\ z(t) &= C_G x_G(t-\tau)\end{aligned} \tag{3.8}$$

then a realization for GQ is given by Theorem 3.5.

Theorem 3.5—State-space realization of GQ
Let Q, and G be given by (3.7) and (3.8). Then a state-space realization of GQ is given by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B\beta r(t)\\ z(t) &= Cx(t-\tau)\end{aligned}$$

where

$$A = \begin{bmatrix} A_G & B_G C_Q \\ 0 & A_Q \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ B_Q \end{bmatrix}, \quad C = \begin{bmatrix} C_G \\ 0 \end{bmatrix}^T$$

for a more complete discussion, less restrictive formulation and proof, consult [Heuberger et al., 2005]. □

We are now equipped to handle all internal closed loop transfer functions.

Transfer functions H_{zn} and H_{ud}

Take $G(s)$ as $P(s)D(s)$ or $P(s)N(s)$ where $D,\ N$ are spectral factors of the load disturbance and measurement disturbance spectral densities respectively. Apply Theorem 3.4 to get

$$\|GQ\|_2^2 = \beta^T B^T M B \beta$$

where M is the solution to

$$A^T M + MA + C^T C = 0$$

Transfer function H_{zd}

$$\begin{aligned} \|H_{zd}\|_2^2 &= \langle P(1-PQ), P(1-PQ)\rangle \\ &= \langle P, P\rangle - \langle P, PPQ\rangle - \langle PPQ, P\rangle + \langle PPQ, PPQ\rangle \end{aligned} \tag{3.9}$$

$\|P\|_2^2$ and $\|PPQ\|_2^2$ can be handled by a state-space realization of PPQ using Theorem 3.5, and solving for the Observability Gramian. The cross terms are handled in the following way: Apply Theorem 3.5, perform a state-space realization of PPQ, then

$$\begin{aligned} \langle P, PPQ\rangle &= B_P e^{\tau A_P^T} M_{P,PPQ} B_{PPQ}\beta \\ \langle PPQ, P\rangle &= \overline{\langle P, PPQ\rangle} \end{aligned}$$

which combined with the previous results for squared norms gives the complete solution.

3.4 State-space realization of the simplified Laguerre filters

This section covers analysis of the simplified Laguerre filters. Expanding Q in the simplified Laguerre filters gives

$$Q(s) = \sum_{i=1}^{N} \frac{\beta_i}{(a+s)^i}$$

which can be realized on Jordan canonical form

$$A_Q = \begin{bmatrix} -a & 1 & 0 & \cdots & 0 \\ 0 & -a & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -a & 1 \\ 0 & 0 & \cdots & 0 & -a \end{bmatrix}, C_Q = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^T \quad B_Q = I_N$$

where I_N is the identity matrix of order N.

Observability Gramian

The Observability Gramian is given by the solution $\mathbf{W}_o$ characterized by

$$\mathbf{W}_o : \quad A^T \mathbf{W}_o + \mathbf{W}_o A = -C^T C$$

which in the case of A only consisting of one Jordan block, and $C = [1,\ 0\ \ldots]$ takes the form

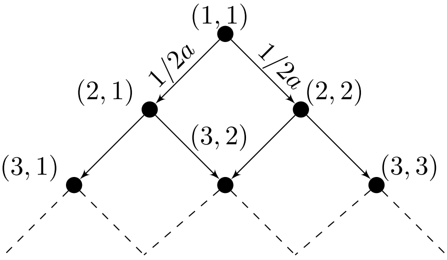


Figure 3.1 A graph representation of the recursion equation (3.11)

$$\begin{bmatrix} -a & & \\ 1 & -a & \\ & \ddots & \ddots \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & \cdots \\ w_{21} & w_{22} & \\ \vdots & & \ddots \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & \cdots \\ w_{21} & w_{22} & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} -a & 1 & \\ & -a & \ddots \\ & & \ddots \end{bmatrix} = \begin{bmatrix} -1 & 0 & \cdots \\ 0 & 0 & \\ \vdots & & \ddots \end{bmatrix} \tag{3.10}$$

This can be expressed as a two-dimensional recursion equation. Let $w_{jk} = 0$ for $j = 0$ or $k = 0$ and the upper left entry of the resulting matrix equation reads $-2aw_{11} = -1 \implies w_{11} = 1/2a$. The Observability Gramian can then be generated from

$$w_{k,j} = \frac{w_{k-1,j} + w_{k,j-1}}{2a} \tag{3.11}$$

(3.11) is illustrated in Figure 3.1 where the weight of each edge is $1/2a$ which is a weighted Pascal's triangle. The closed form solution of the Observability Gramian is given by

$$w_{j,k} = \frac{(j+k-2)!}{(j-1)!(k-1)!} \left(\frac{1}{2a}\right)^{j+k-1} \tag{3.12}$$

which we can write on matrix form as

$$\mathbf{W}_o = \text{diag}\begin{bmatrix}1 & (1/2a)^{1/2} & (1/2a)^{3/2} & \cdots\end{bmatrix} \cdot P_S \cdot \text{diag}\begin{bmatrix}1 & (1/2a)^{1/2} & (1/2a)^{3/2} & \cdots\end{bmatrix} \tag{3.13}$$

where P_S is a symmetric Pascal matrix of order N. The Pascal matrix has scaling issues; the condition number of the Pascal matrix grows exponentially and MATLAB has difficulties treating such a matrix, see Figure 3.2.

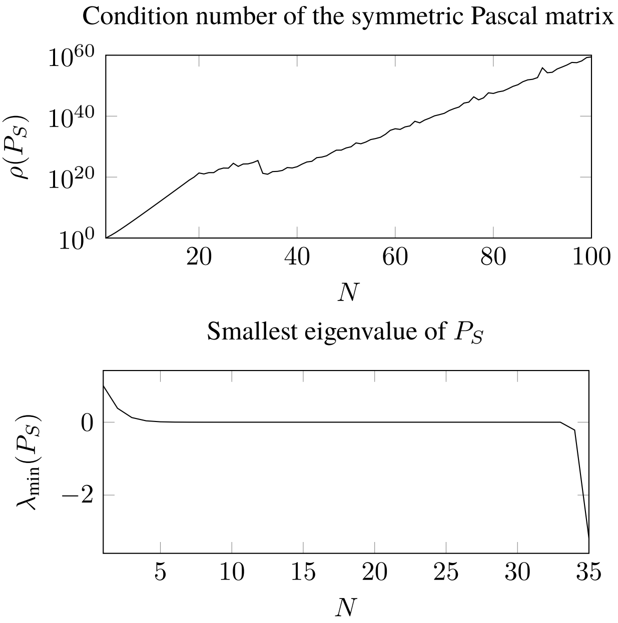


Figure 3.2 Condition number and minimum eigenvalue of the symmetric Pascal matrix computed using `cond` and `eig` in MATLAB.

As we can see in Figure 3.2 MATLAB cannot handle the simplest of costs ($\|H_{un}\|_2^2 = \|Q\|_2^2$) properly. Motivates the use of other basis functions.

3.5 State-space realization of the Laguerre filters

One basis that should not have the inherent poor scaling of the simplified Laguerre filters are the orthonormal Laguerre filters. Since all cross terms are zero, we know that $\|Q\|_2^2 = \beta^T\beta$. Since

$$Q = \sum_{k=0}^{N-1} q_k(s)\beta_k$$

consider

$$
\begin{aligned}
q_k(s) &= \frac{\sqrt{2a}}{a+s}\left(\frac{a-s}{a+s}\right)^k \\
&= (-1)^k\left(1-\frac{2a}{s+a}\right)^k\frac{\sqrt{2a}}{s+a} \\
&= \frac{(-1)^k\sqrt{2a}}{s+a}\Bigg(\binom{k}{0}+\binom{k}{1}\frac{-2a}{s+a}+\cdots \\
&\qquad +\binom{k}{k-1}\left(\frac{-2a}{s+a}\right)^{k-1}+\binom{k}{k}\left(\frac{-2a}{s+a}\right)^k\Bigg)
\end{aligned}
$$

thus we get

$$
\begin{aligned}
\sum_{k=0}^{N-1} q_k(s)b_k = \sqrt{2a}\Bigg(&\frac{1}{s+a}\sum_{k=0}^{N-1}\binom{k}{0}(-1)^k b_k \\
&+\frac{1}{(s+a)^2}\sum_{k=1}^{N-1}\binom{k}{1}(-1)^k(-2a)b_k\Bigg)+\cdots \\
&+\sqrt{2a}\left(\frac{1}{(s+a)^{N-1}}\sum_{k=N-2}^{N-1}\binom{k}{N-2}(-1)^k(-2a)^{N-2}b_k\right) \\
&+\sqrt{2a}\left(\frac{1}{(s+a)^{N}}\sum_{k=N-1}^{N-1}\binom{k}{N-1}(-1)^k(-2a)^{N-1}b_k\right)
\end{aligned}
$$

which, by collecting all the terms associated with each pole, is realizable on a Jordan form as

$$
A = \begin{bmatrix} -a & 1 & 0 & \cdots & 0 \\ 0 & -a & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -a & 1 \\ 0 & 0 & \cdots & 0 & -a \end{bmatrix}
$$

$$
B = \sqrt{2a}M_a \underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & \cdots \\ 0 & 1 & -2 & 3 & -4 & \cdots \\ 0 & 0 & 1 & -3 & 6 & \cdots \\ 0 & 0 & 0 & 1 & -4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \cdots \end{bmatrix}}_{P_U^{-1}}
$$

$$
C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}
$$

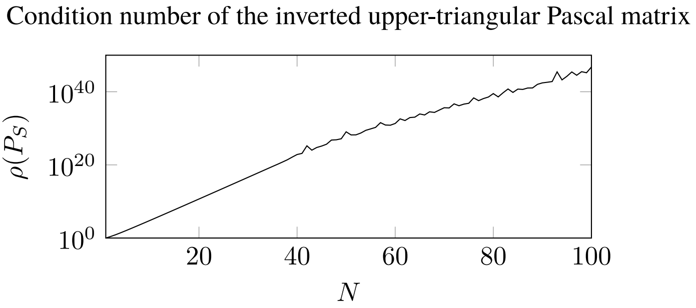


Figure 3.3 Condition number of the inverted upper triangular Pascal matrix, calculated using MATLAB's cond.

where

$$M_a = \text{diag}\left[1\ 2a\ (2a)^2\ \cdots\ (2a)^{N-1}\right]$$

and $P_U = P_L^T$. This realization has poor numerical properties when it comes to calculating norms. The reason for this is that

$$\|P(s)Q(s)\| = \beta^T B^T \mathbf{W}_o B\beta$$

requires right and left multiplication by $B = [0\ B_q^T]^T$. The elements grow quickly and the condition number grows exponentially, see Figure 3.3.

3.6 Output-orthogonal realization of the Laguerre filters

We will now explore what happens if we apply the transform $T\hat{x} = x$, where $T = B$. This leads to the system

$$\begin{aligned}\dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}\beta u\\ y &= \hat{C}\hat{x}\end{aligned}$$

where

$$\begin{aligned}\hat{A} &= T^{-1}AT\\ \hat{B} &= T^{-1}B = \mathbf{I}\\ \hat{C} &= CT\end{aligned}$$

Since the aim is to avoid poorly conditioned matrix multiplications we need to solve for $\hat{A},\ \hat{C}$ analytically. Matrix multiplications yields

$$X = M_a^{-1}AM_a = \begin{bmatrix} -a & 2a & 0 & \cdots \\ 0 & -a & 2a & \ddots \\ 0 & 0 & -a & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

partition X as $X = -aI + 2aI_{+1}$, where I_{+1} is a matrix with ones on the super diagonal, zeros elsewhere. Then

$$P_U I_{+1} P_U^{-1} = \begin{bmatrix} 0 & 1 & -1 & 1 & \cdots \\ 0 & 0 & 1 & -1 & \ddots \\ 0 & 0 & 0 & 1 & \ddots \\ 0 & 0 & 0 & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

This is an upper-triangular Toeplitz matrix with zeros on the diagonal and 1 on the odd diagonals and -1 on the even diagonals (starting at 0).

Combining this gives

$$\hat{A} = \begin{bmatrix} -a & 2a & -2a & 2a & \cdots \\ 0 & -a & 2a & -2a & \ddots \\ 0 & 0 & -a & 2a & \ddots \\ 0 & 0 & 0 & -a & \ddots \\ \cdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

and

$$\hat{C} = CT = \sqrt{2a}[1 \ -1 \ 1 \ \cdots]$$

These results are summarized in Definition 3.1.

DEFINITION 3.1—OUTPUT-ORTHOGONAL REALIZATION OF LAGUERRE BASIS

Let $Q(s) = \sum_{i=1}^{N} \beta_i q_i(s)$ where $q_i(s)$ are the Laguerre filters of order i defined in 2.10. Then the output-orthogonal state-space realization is given by

$$A = \begin{bmatrix} -a & 2a & -2a & \\ & -a & 2a & \ddots \\ & & -a & \ddots \\ & & & \ddots \end{bmatrix}, \ C = \sqrt{2a} \begin{bmatrix} 1 \\ -1 \\ 1 \\ \vdots \end{bmatrix}^T, \ B = I \tag{3.14}$$

□

Measures of accuracy We know from Lemma 3.3 that the resulting Hessians will be Toeplitz. State-space methods does not enforce the Toeplitz structure, thus it is possible to introduce a measure to get a lower bound on the magnitude of the error.

DEFINITION 3.2—DISTANCE TO TOEPLITZ STRUCTURE

The distance from a square matrix M to Toeplitz structure, in Frobienius norm is given by

$$d_{\text{toeplitz}}(M) = ||M - T_M||_F$$

where T_M is the Toeplitz matrix that minimizes $\|M - T_M\|_F$, and $\|\cdot\|_F$ is the Frobenius norm. □

The distance to Toeplitz structure has a closed form expression and is presented in the following Theorem.

Theorem 3.6—Distance to Toeplitz structure
The distance to Toeplitz structure as defined in Definition 3.2 is given by

$$d_{\text{toeplitz}}(M) = \sum_{i=1}^{N}(m_{i,i} - t_0)^2 + \sum_{s=1}^{N-1}\sum_{p=1}^{N-s}\left((m_{p,p+s} - t_s)^2 + (m_{p+s,p} - t_{-s})^2\right)$$

where $m_{i,j}$ is the element on the ith row and the jth column of M, and t_s is given by

$$t_s = \begin{cases} \displaystyle\sum_{p=1}^{N-s} \frac{m_{p,p+s}}{N-s}, & s > 0 \\ \displaystyle\sum_{p=1}^{N-s} \frac{m_{p+s,p}}{N-s}, & s < 0 \end{cases}$$

Proof Let $\mathcal{T}$ be the set of Toeplitz matrices, and $T_M \in \mathcal{T}$ be of equal size to a give matrix M then

$$\begin{aligned} d_{\text{toeplitz}}(M) &= \min_{T_M \in \mathcal{T}} \|M - T_M\|_F \\ \|M - T_M\|_F^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N}(m_{i,j} - t_{j-i})^2 \\ &= \sum_{i=1}^{N}(m_{i,i} - t_0)^2 \\ &\quad + \sum_{s=1}^{N-1}\sum_{p=1}^{N-s}\left((m_{p,p+s} - t_s)^2 + (m_{p+s,p} - t_{-s})^2\right) \end{aligned}$$

where t_s is the element associated with the sth diagonal of T_M. Note that the problem is separable in s (we can solve for each s separately)

$$\sum_{i=1}^{N}(m_{i,i} - t_0)^2 = \sum_{i=1}^{N} m_{i,i}^2 - \left(\sum_{i=1}^{N} m_{i,i}\right)^2 + \left(Nt_0 - \sum_{i=1}^{N} m_{i,i}\right)^2$$

which has it's minimum when

$$t_0 = \frac{\sum_{i=1}^{N} m_{i,i}}{N}$$

Similarly we can solve for $s = -N, \dots, N$

$$t_s = \begin{cases} \displaystyle\sum_{p=1}^{N-s} \frac{m_{p,p+s}}{N-s}, & s > 0 \\ \displaystyle\sum_{p=1}^{N-s} \frac{m_{p+s,p}}{N-s}, & s < 0 \end{cases}$$

□

Theorem 3.6 is can be used to debug code, and to compare algorithms. An intuitive interpretation is that the closest Toeplitz matrix, in Frobenius norm, is constructed by averaging each diagonal.

3.7 Lyapunov equations exploiting Toeplitz structure

In this section we explore whether we can exploit the Toeplitz structure to arrive at an improved algorithm for solving the Lyapunov equations.

Partitioning

Consider the Lyapunov equation

$$\begin{bmatrix} A_p^T & 0 \\ (B_pC_q)^T & A_q^T \end{bmatrix} \begin{bmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{bmatrix} + \begin{bmatrix} X_1 & X_2 \\ X_2^T & X_3 \end{bmatrix} \begin{bmatrix} A_p & B_pC_q \\ 0 & A_q \end{bmatrix} = - \begin{bmatrix} C_p^T C_p & 0 \\ 0 & 0 \end{bmatrix}$$

where X_3 is known to be Toeplitz from lemma 3.3, and $A_q,\ C_q$ are given by (3.14). Partition X to get three equations

$$A_p^T X_1 + X_1 A_p = -C_p^T C_p \tag{3.15a}$$

$$A_p^T X_2 + X_2 A_q = -X_1 B_p C_q \tag{3.15b}$$

$$\begin{aligned} A_q^T X_3 + X_3 A_q &= -(B_pC_q)^T X_2 - X_2^T (B_pC_q) \\ &= Y \end{aligned} \tag{3.15c}$$

where (3.15a) gives the Observability Gramian of the plant, (3.15b) is a Sylvester equation. If the output-orthogonal realization of the Laguerre filters are used (3.15c) gives the Hessian of the QP. X_3 is the Hessian of the QP. This can be seen from

$$M = \begin{bmatrix} 0 & I_N \end{bmatrix} \begin{bmatrix} X_0 & X_2 \\ X_2^T & X_3 \end{bmatrix} \begin{bmatrix} 0 \\ I_N \end{bmatrix} = X_3$$

Consider (3.15c), where A_q is given by (3.14)

$$\begin{bmatrix} -a & & & \\ 2a & -a & & \\ -2a & -2a & -a & \\ & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots \\ x_{12} & x_{22} & \\ \vdots & & \ddots \end{bmatrix}$$

$$+ \begin{bmatrix} x_{11} & x_{12} & \cdots \\ x_{12} & x_{22} & \\ \vdots & & \ddots \end{bmatrix} \begin{bmatrix} -a & 2a & -2a & \\ & -a & 2a & \ddots \\ & & -a & \ddots \\ & & & \ddots \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots \\ y_{12} & y_{22} & \\ \vdots & & \ddots \end{bmatrix} \tag{3.16}$$

Since X_3 is Toeplitz it's enough to solve the first column (or row) of X_3. Note that $x_{11} = -\frac{y_{11}}{2a}$. Then

$$\begin{aligned} x_{12} - x_{11} &= -\frac{y_{12}}{2a} \\ x_{13} - x_{12} + x_{11} &= -\frac{y_{13}}{2a} \\ &\cdots \\ x_{1N} - x_{1(N-1)} + \cdots &= -\frac{y_{1N}}{2a} \end{aligned}$$

By adding the previous row from the current row this relation can be rewritten as

$$x_{1k} = -\frac{y_{1k} + y_{1(k-1)}}{2a}$$

which gives us an expression for the first row (column) of X_3 and therefore, as X_3 is a symmetric Toeplitz matrix, an expression for the whole matrix.

3.8 Comparison — numerical example

The aim of this section is to compare three algorithms of computing the Hessian (M) when Q has been expanded using Laguerre filters. The system is the same as in Example 1.1

$$\begin{aligned} P(s) &= \frac{1}{(s+1)(s^2 + 2 \cdot 0.01s + 1)} \\ D(s) &= \frac{1}{s/0.1 + 1} \\ N(s) &= 1 \end{aligned}$$

where $M \in \{M | \ \beta^T M \beta = \|H_{ud} D\| = \|PDQ\|, \ \forall \beta\}$ is to be calculated. The algorithms used are:

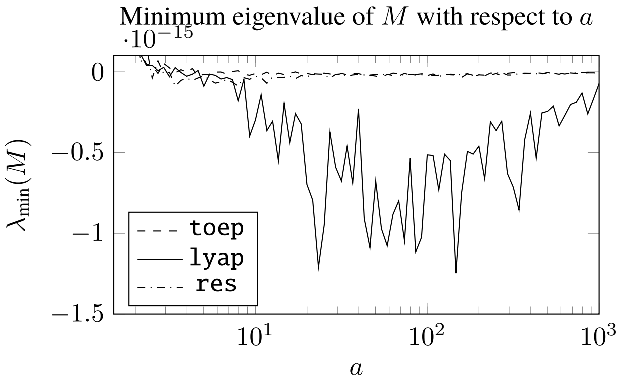


Figure 3.4 Value of the smallest eigenvalue, calculated using `eig` of the different algorithms for different values of a

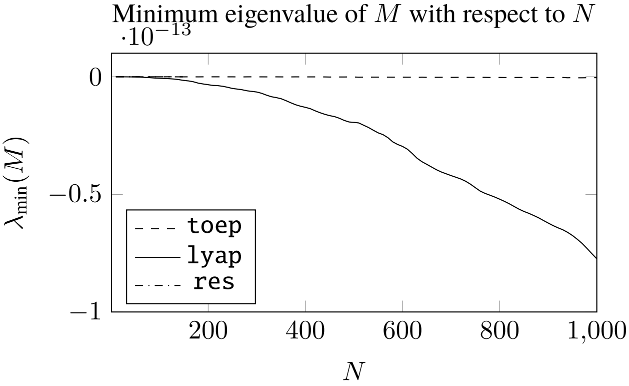


Figure 3.5 Value of the smallest eigenvalue, calculated using `eig` of the different algorithms for different values of N when $a = 100$, for $N > 150$ *Residues* explodes and some elements in M approach infinity.

1. *Residues*: Based on calculus of residues, using the limit theorem and (2.5).
2. *Lyap*: Based on the output-orthogonal realization of the Laguerre filters using `lyap` to calculate M.
3. *Toep*: Based on the output-orthogonal realization of the Laguerre filters, calculating M using the method described in section 3.7.

Sweeping for different values of a and N results in the following The results of sweeping for different values of a and N are shown in Figures 3.4, 3.5 and 3.6.

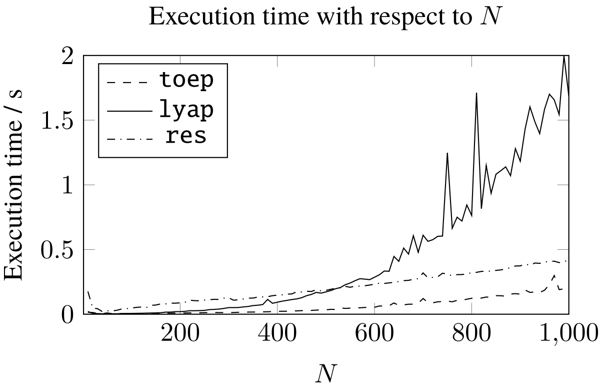


Figure 3.6 Execution time of the three different algorithms run on a Dell Latitude E7240

Observe that by exploiting the Toeplitz substructure we are able to get faster and in this case more accurate algorithms. The algorithm based on calculus of residues fails to produce solutions for $N > 150$ where some elements become infinite. The figures also highlight the issue that M becomes indefinite for all algorithms. But by Lemma 3.2, it must be positive semidefinite, thus there has been numerical errors. Secondly the problem is a convex optimization problem only if M is positive (semi) definite. The last issue can be handled by using the Hessian $M + \gamma I_N$, where γ is a positive real scalar larger in absolute value than the smallest eigenvalue of M. A larger in magnitude, negative eigenvalue requires a larger modification of the cost function, thus decreasing accuracy of the solution.

4

Extension to MIMO systems

This chapter explores closed loop transfer function matrices of stable systems on the form

$$H_{zw}(i\omega) = P_{zw}(i\omega) - P_{zu}(i\omega)Q(i\omega)P_{yw}(i\omega) \tag{4.1}$$

Where $\dim(P_{zw}) = n_z \times n_w$, $\dim(P_{zu}) = n_z \times n_u$, $\dim(Q) = n_u \times n_y$, and $\dim(P_{yw}) = n_y \times n_w$. Section 4.1 show that by representing the transfer function in vector form, we can avoid the post multiplication of the matrix P_{yw} and store all the coefficients in a vector. Section 4.2 shows how to transform quadratic cost functionals on the subspace of $\mathcal{H}_2$ spanned by the Ritz approximation of Q to the finite-dimension space of coefficient vectors.

4.1 Vectorizing the transfer Matrix

Preliminaries

Before diving into the transfer matrix we will explore the vector operator and related algebra. We will start by defining the vector operator and the Kronecker product, followed by the necessary relations.

Definition 4.1—Vector operator
Let A be an $m \times n$ matrix, then $\text{vec}(A)$ is given by

$$\text{vec}(A) = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} & a_{12} & \cdots & a_{mn} \end{bmatrix}^T \qquad \square$$

Vectorizing a matrix can be thought of as stacking the columns on top of each other, such that the leftmost column is on top, and the rightmost column is at the bottom of the vector.

Definition 4.2—Kronecker product
Let A and B be matrices of dimensions $m \times n$ and $p \times q$ respectively. The Kronecker

product between A and B are then given by

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \qquad \square$$

Some important properties of the vector operator are

$$\operatorname{vec}(\alpha D + \beta E) = \alpha \operatorname{vec}(D) + \beta \operatorname{vec}(E) \tag{4.2a}$$

$$a \otimes b = \operatorname{vec}(ba^T) \tag{4.2b}$$

$$\operatorname{vec}(AB) = (B^T \otimes I_m)\operatorname{vec}(A) \tag{4.2c}$$

$$\operatorname{vec}(ABC) = (C^T \otimes A)\operatorname{vec}(B) \tag{4.2d}$$

$$\operatorname{tr}\big(A^T B\big) = \operatorname{vec}(A)^T \operatorname{vec}(B) \tag{4.2e}$$

where A, B, C, D, E are matrices of appropriate sizes. a and b are column vectors, α and β are scalars.

The transfer matrix

So far we have assumed Q to be scalar. For MIMO systems Q is a matrix, of dimension $n_u \times n_y$ where n_u is the dimension of the control signal and n_y is the dimension of the sensor signal. By using the same set of basis functions for each element in Q we get

$$Q(i\omega) = \begin{bmatrix} Q_{11}(i\omega) & \cdots & Q_{1n_y}(i\omega) \\ \vdots & \ddots & \vdots \\ Q_{n_u1}(i\omega) & \cdots & Q_{n_un_y}(i\omega) \end{bmatrix}$$

where $Q_{ij} = \sum_k \beta_{ij}^k q_k$. Writing Q in vector form gives

$$\begin{aligned} \operatorname{vec}(Q) &= \underbrace{\begin{bmatrix} \beta_{11}^1 & \cdots & \beta_{11}^N \\ \vdots & \vdots & \vdots \\ \beta_{n_u1}^1 & \cdots & \beta_{n_u1}^N \\ \vdots & \vdots & \vdots \\ \beta_{n_un_y}^1 & \cdots & \beta_{n_un_y}^N \end{bmatrix}}_{\mathcal{B}} \underbrace{\begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix}}_{q} \\ &= (q^T \otimes I_{n_yn_u})\, \underbrace{\operatorname{vec}(\mathcal{B})}_{\beta} \end{aligned}$$

where the last line was gotten by applying (4.2c). Applying (4.2d) and (4.2a) to (4.1) gives

$$\begin{aligned}\mathrm{vec}(H_{zw}) &= \mathrm{vec}(P_{zw}) - (P_{yw}^T \otimes P_{zu})\mathrm{vec}(Q) \\ &= \mathrm{vec}(H_{zw}) = \underbrace{\mathrm{vec}(P_{zw})}_{T_0} - \underbrace{(P_{yw}^T \otimes P_{zu})(q^T \otimes I_{n_y n_u})}_{T_1}\beta \end{aligned} \tag{4.3}$$

4.2 Constructing the cost function

Consider again the cost functional (3.1). This can be written i terms of inner products as

$$J_{\text{lqg}} = \langle \mathrm{vec}(H_{zw}), V\mathrm{vec}(H_{zw})\rangle \tag{4.4}$$

where V is a positive semidefinite weight matrix. To recover (3.1) we may pick

$$V = \begin{bmatrix} a & & & \\ & b & & \\ & & c & \\ & & & d \end{bmatrix}$$

To represent a larger class of cost functionals V may be chosen arbitrarily among the set of $n_z n_w \times n_z n_w$ positive semidefinite matrices. Let T_0 and T_1 have the state-space realizations.

$$\begin{aligned} T_0 \quad \dot{x} &= A_0 x + B_0 u \\ y &= C_0 x \\ T_1 \quad \dot{x} &= A_1 x + B_1 u \\ y &= C_1 x \end{aligned}$$

Inserting (4.3) into (4.4)

$$\begin{aligned} J &= \langle T_0 - T_1\beta, V(T_0 - T_1\beta)\rangle \\ &= \langle T_0, VT_0\rangle - 2\langle T_0, VT_1\beta\rangle + \langle T_1\beta, VT_1\beta\rangle \end{aligned}$$

Applying Theorem 3.4 to the equation results in

$$J = (1/2)\beta^T M\beta + q_0^T\beta + r_0$$

where $M = 2B_1^T X B_1$, $q_0 = -2B_0 Y B_1$ and $r_0 = B_0^T Z B_0$. X, Y and Z are given by the solutions to the following Sylvester and two Lyapunov equations

$$A_1^T X + XA_1 + C_1^T V C_1 = 0 \tag{4.5a}$$

$$A_0^T Y + YA_1 + C_0^T V C_1 = 0 \tag{4.5b}$$

$$A_0^T Z + ZA_0 + C_0^T V C_0 = 0 \tag{4.5c}$$

A word of caution, we recommend solving (4.5b) using `lyap` and not `sylv` in MATLAB. The reason is that `sylv` is poorly implemented. The next example is intended to clarify how T_0 and T_1 can be constructed.

EXAMPLE 4.1—RECOVER LQG USING GENERAL PLANT DESCRIPTION

Consider the 1-DOF case illustrated in Figure 1.1. Take $z = \begin{bmatrix} y_{\text{out}} & u \end{bmatrix}^T$, y = $-y_{\text{sensor}} - n$ and $w = \begin{bmatrix} d & n \end{bmatrix}$. Let $N(s)$, $D(s)$ be the spectral factors of measurement noise and process disturbance and W_n, W_d be their respective intensities. Taking $Q = q^T \beta$ we get

$$\text{vec}(H_{zw}) = \begin{bmatrix} H_{y_{\text{out}}d} \\ H_{ud} \\ H_{y_{\text{out}}n} \\ H_{un} \end{bmatrix} = \underbrace{\begin{bmatrix} P(s)D(s) \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{T_0} - \underbrace{\begin{bmatrix} W_d P(s)D(s)P(s)q^T(s) \\ -W_d D(s)P(s)q^T(s) \\ -W_n N(s)P(s)q^T(s) \\ -W_n N(s)q^T(s) \end{bmatrix}}_{T_1} \beta$$

T_1 can then be realized as

$$T_1 : \quad \begin{aligned} A_1 &= \begin{bmatrix} A_p & B_pC_p & 0 & 0 & 0 & 0 \\ 0 & A_p & 0 & B_pC_d & 0 & 0 \\ 0 & 0 & A_p & 0 & B_pC_n & 0 \\ 0 & 0 & 0 & A_d & 0 & B_dC_q \\ 0 & 0 & 0 & 0 & A_n & B_nC_q \\ 0 & 0 & 0 & 0 & 0 & A_q \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ B_q \end{bmatrix} \\ C_1 &= \begin{bmatrix} W_dC_p & 0 & 0 & 0 & 0 & 0 \\ 0 & -W_dC_p & 0 & 0 & 0 & 0 \\ 0 & 0 & -W_nC_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -W_nC_n & 0 \end{bmatrix} \end{aligned}$$

and T_0 as

$$T_0 : \qquad A_0 = A_{pd}, \quad B_0 = B_{pd}, \quad C_0 = \begin{bmatrix} W_dC_{pd} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

□

5

Results

5.1 MIGO test-batch

In this section we present the results of applying four algorithms to recover an LQG controller with the cost functional (2.8) on the MIGO test-batch benchmark systems. The algorithms considered all use the output orthogonal realization, see Section 3.6.

- `4lyap`: Theorem 3.4 is applied to (2.8). M is obtained by solving 4 Lyapunov equations, q_0 is obtained by solving a Sylvester equation, and r_0 is obtained by solving a Lyapunov equation, all using `lyap`. M is then factored using `chol` and the problem is rewritten as a SOCP and solved using MOSEK.

- `toep`: Similar to `4lyap`, with the exception that the 4 Lyapunov equations used to obtain M are solved using the method described in Section 3.7.

- `1lyap`: Based on Chapter 4. The Sylvester and two Lyapunov equations are solved using `lyap`. The resulting QP is solved directly using MOSEK.

- `chol`: Also based on Chapter 4 but differs from `1lyap` in that `lyapchol` is used to solve the Lyapunov equation to obtain R such that R is a Cholesky factor of M and that a SOCP was solved using MOSEK.

System description

This is a subset of a large set of systems often encountered in the Process industry. The set of systems first appeared in [Hägglund and Åström, 2004]. We consider the following systems

$$
\begin{aligned}
P_1(s) &= \frac{e^{-s}}{1+sT}, \\
P_2(s) &= \frac{e^{-s}}{(1+sT)^2}, \\
P_3(s) &= \frac{1}{(s+1)^n} \\
P_4(s) &= \frac{s-\alpha}{(s+1)^3}
\end{aligned}
$$

Recover LQG

Problem 1 Recover the LQG for the system.

$$
\begin{aligned}
P_1(s) &= \left(\frac{s^2-6s+12}{s^2+6s+12}\right)\frac{1}{1+0.05s} \\
D(s) &= \frac{1}{1+0.5s} \\
N(s) &= 1
\end{aligned}
$$

where the intensity of the process disturbance, and measurement noise are given by $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively and $\rho = 0.3$. Note that the time delay of P_1 in the MIGO test-batch has been approximated using a second-order Padé approximant.

Note in Figure 5.1 that we are able to recover the LQG cost using `4lyap`, `1lyap` and `toep` but that `chol` gives numerical issues for higher frequencies. Also note that the convergence rate is highly dependent on a.

Problem 2 Recover the LQG for the system

$$
\begin{aligned}
P_2(s) &= \left(\frac{s^2-6s+12}{s^2+6s+12}\right)\frac{1}{(1+sT)^2} \\
D(s) &= \frac{1}{1+10sT} \\
N(s) &= 1
\end{aligned}
$$

where $T = [0.02,\ 1000]$. A lower value of T results in a fast system, whereas a increasing the value of T results in a slower system. The intensities of the process disturbance and measurement noise are given by $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively, and $\rho = 0.3$. Just as in the previous problem the time delay of P_2 in the MIGO test-batch has been approximated using a second-order Padé approximant.

Note from Figures 5.2 and 5.3 that the rate of convergence with respect to N is highly dependent on the cutoff frequency a of the filters. The range of a for which the rate of convergence is satisfactory seems to be dependent on the bandwidth of

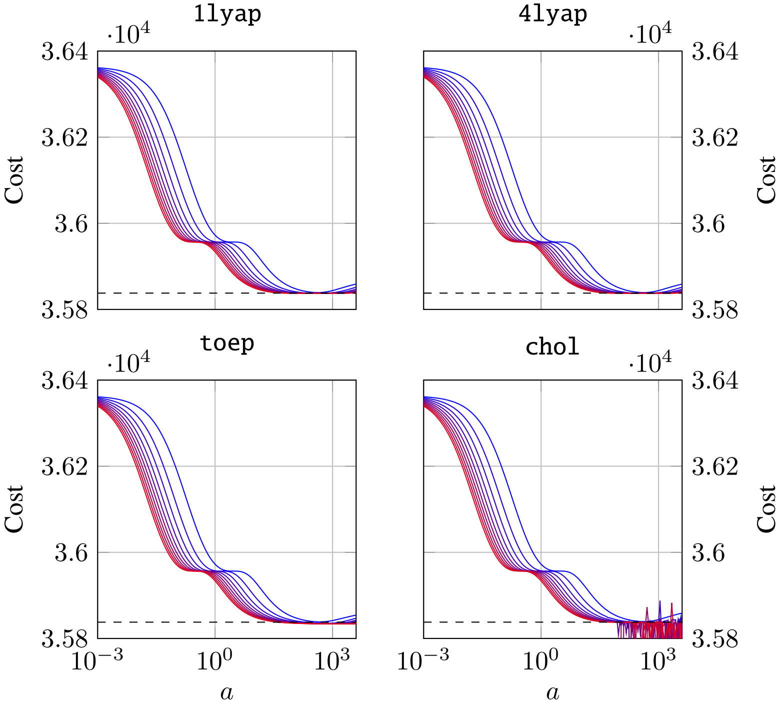


Figure 5.1 The results of solving P_1 for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

the system. Also note that `4lyap` and `1lyap` are the only algorithms that gives a satisfactory result for the faster system, and that `chol` works poorly for the faster system. `toep` also fails to deliver reasonable results as the cost in Figure 5.2 is lower than optimum.

Problem 3 Recreate the LQG for the system

$$
\begin{aligned}
P_3(s) &= \frac{1}{(s+1)^n} \\
D(s) &= \frac{1}{s+10s} \\
N(s) &= 1
\end{aligned}
$$

for $n = 6,\ 8$. The process disturbance and measurement noise intensities are given by $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively, and $\rho = 0.3$.

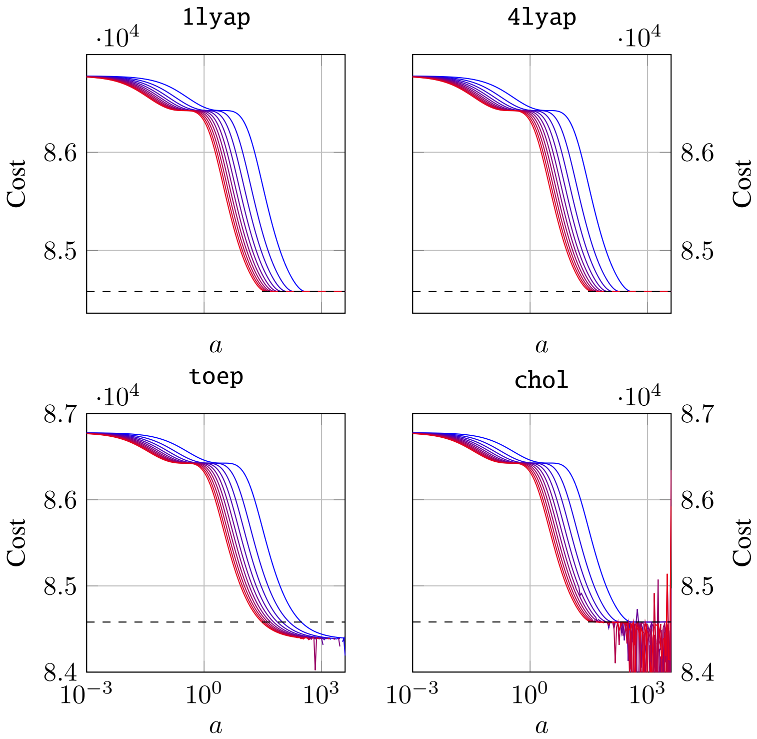


Figure 5.2 The results of solving P_2 where $T = 0.02$ for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

From Figures 5.4 and 5.5 we conclude that the increased multiplicity of the pole leads to a smaller range of a where optimum is achieved and an increase in plateaus. We can't observe any difference between `1lyap` and `4lyap`. We experience a small amount of numerical difficulties using `chol` when $n = 6$, but not for $n = 8$. However `toep` does not provide a satisfactory result.

Problem 4 Recreate the LQG for the system:

$$
\begin{aligned}
P_4(s) &= \frac{s - 0.1}{(s+1)^3} \\
D(s) &= \frac{1}{1 + 10s} \\
N(s) &= 1
\end{aligned}
$$

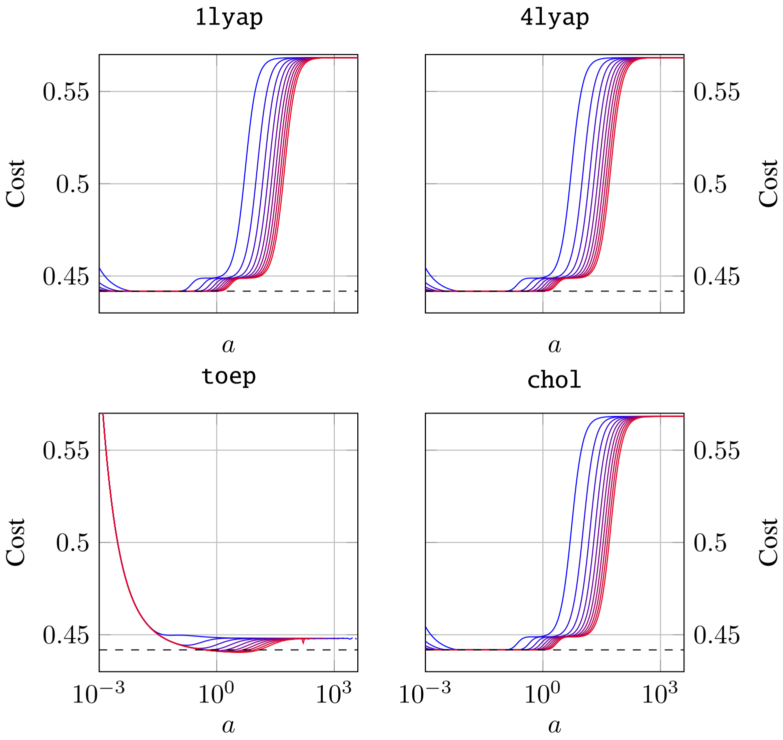


Figure 5.3 The results of solving P_2 where $T = 1000$ for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

where the intensities of the process disturbance and measurement noise are $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively, and $\rho = 0.3$.

This problem was solved only using `4lyap` and `1lyap`. Especially note the slow convergence, the importance of selecting the cutoff frequency properly and that the a which leads to the lowest cost for $N = 10$ does not equal the a which leads to the lowest cost for $N \geq 20$.

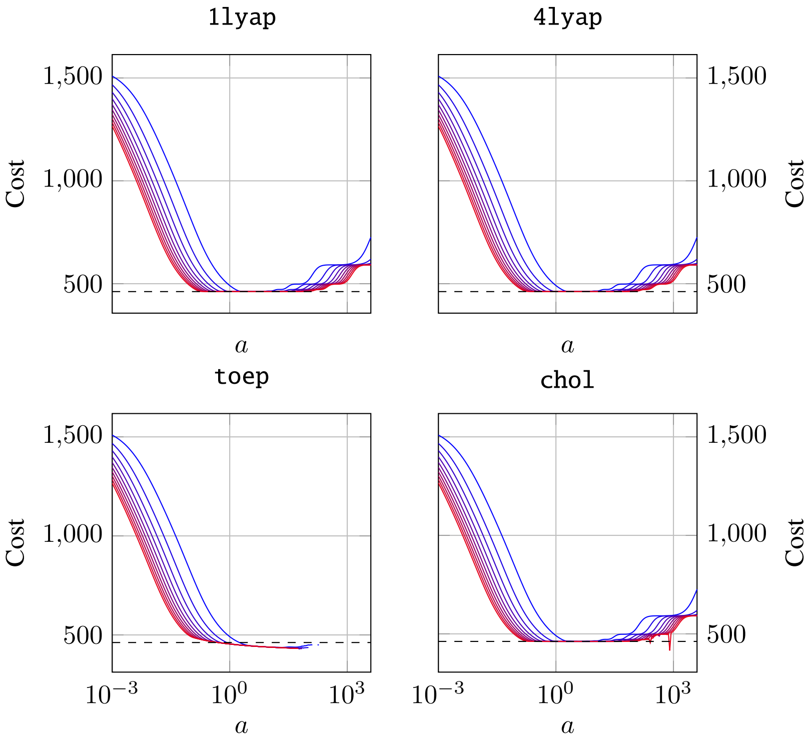


Figure 5.4 The results of solving P_3 where $n = 6$ for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

Exact solution for systems with time delay

An interesting implication of Theorem 3.4 is that time delay only appears in the Sylvester equation used to obtain q_0. Using `expm` to calculate the matrix exponential in algorithm `4lyap` allows us to solve for optimal regulators of systems with time-delay exactly.

Problem 1 revisited Consider again the same problem as problem one but without Padé approximant:

$$P_1(s) = \frac{e^{-s}}{1 + 0.05s}$$

$$D(s) = \frac{1}{1 + 0.5s}$$

$$N(s) = 1$$

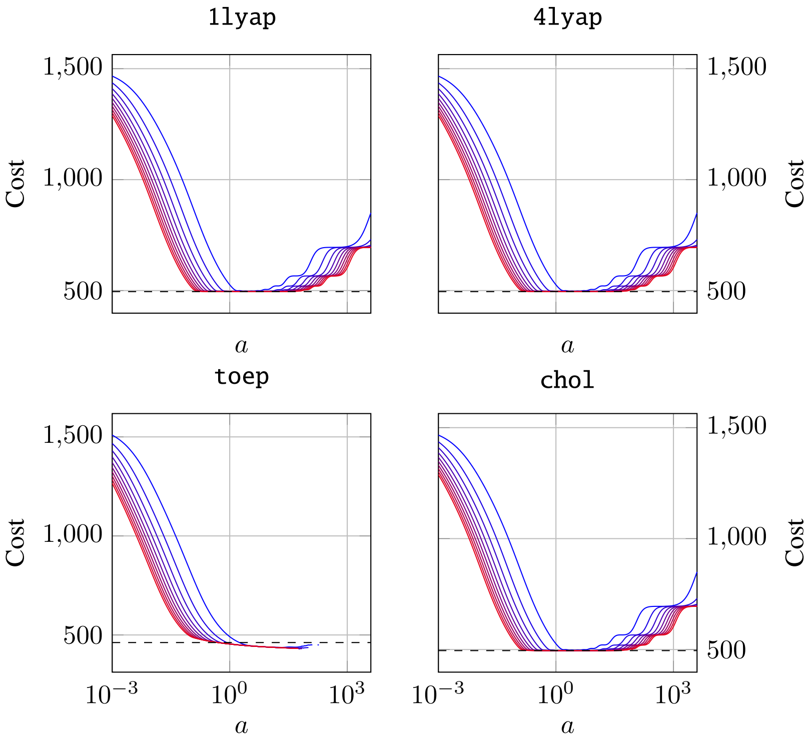


Figure 5.5 The results of solving P_2 where $n = 8$ for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

where the intensities of the process disturbance and measurement noise are $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively, and $\rho = 0.3$. The LQG has been calculated by replacing the time-delay in P_1 by a first-order Padé approximant. The results of solving P_1 using 4lyap is plotted together with the cost of using the LQG controller in Figure 5.7. Observe that the cost is lower using 4lyap than benchmark LQG and that the maximum sensitivity is much lower, even though no constraint on maximum sensitivity was imposed. Also note the importance of properly selecting a.

The minimum in Figure 5.7 can be reached by designing an LQG regulator and approximating the time-delay by a 6th-order Padé approximant. The cost does not decrease with approximants of higher order.

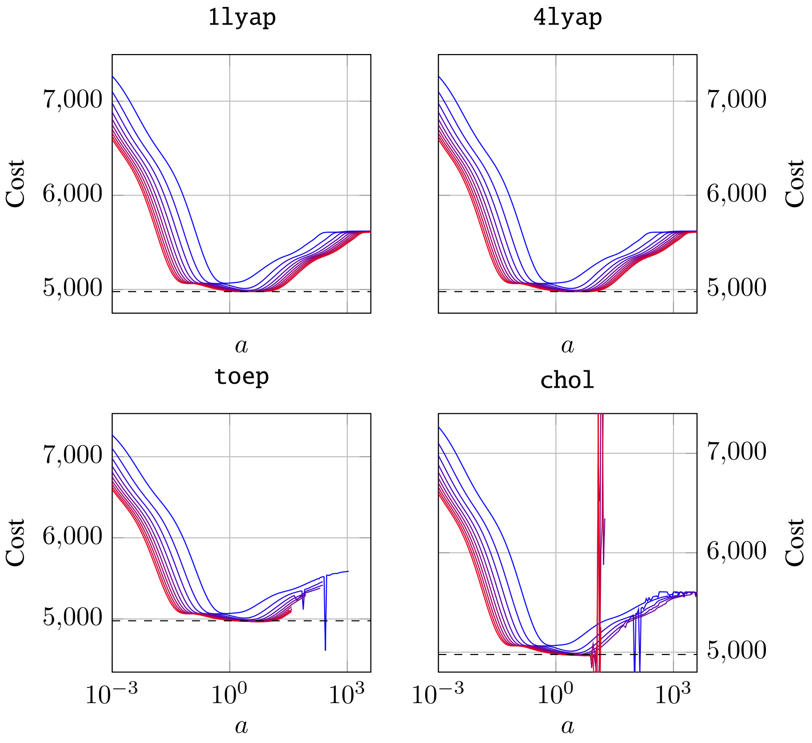


Figure 5.6 The results of solving P_4 for different values of a and N using four different algorithms. N ranges from 10 (blue) to 100 (red) in steps of 10. The dashed line represent the true minimum, obtained by calculating the cost of using the LQG controller.

5.2 Effects of regularization

Consider the following system which was deliberately constructed to make `4lyap` fail

$$
\begin{aligned}
P_f(s) &= \prod_{i=1}^{2} \left(\frac{s-a_i}{s+a_i} \right) \frac{(\omega_i^p)^2}{s^2 + 2\zeta\omega_i^p s + (\omega_i^p)^2} \\
D(s) &= \frac{1}{100s+1} \prod_{i=1}^{2} \frac{(\omega_i^d)^2}{s^2 + 2\zeta\omega_i^d s + (\omega_i^d)^2} \\
N(s) &= 1
\end{aligned}
$$

where $a = [0.1,\ 1]$, $\omega^p = [0.2,\ 7]$, $\omega^d = [0.1,\ 0.2]$. The process and noise intensity are given by $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$ respectively. The result of running

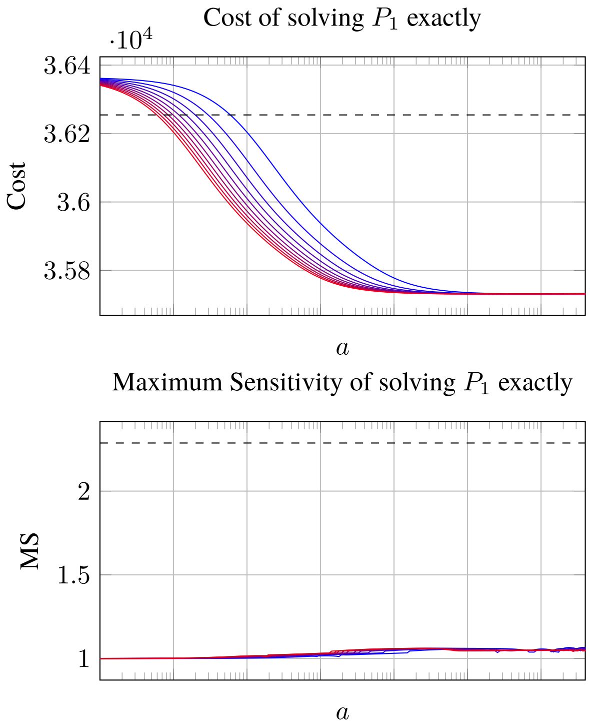


Figure 5.7 Solving P_1 exactly for different values of a and N, where N ranges from 10 (blue) to 100 red.

`4lyap` can be seen in the left plot of Figure 5.8. This result is not satisfactory and it comes from the Hessian becoming indefinite for frequencies higher than $a \approx 0.05$. For example setting $N = 70,\ a = 1$ gives the minimum eigenvalue $\lambda_{\text{min}} = -0.5$ and the condition number $\rho(M) = 3.4 \cdot 10^{17}$. This means that the minimum eigenvalue is zero within working precision, so we might try to force the Hessian to become positive definite, such that the minimum eigenvalue is still zero within working precision. We can do this by using $\hat{M} = M - \gamma \min(\lambda) I_N$, where $\gamma = 2$ if the minimum eigenvalue is negative, else 0. The results are shown in the right plot of Figure 5.8. Note that we do reach optimum, and that the numerical issues seems to be somewhat suppressed, but that the relation between cost and a is less smooth.

5.3 Flatbed test-case

The plant is a flexible structure used as a damage mitigation flatbed [Sznaier, 2000]. The structure consists of two masses, whose position are y_1 and y_2, supported by

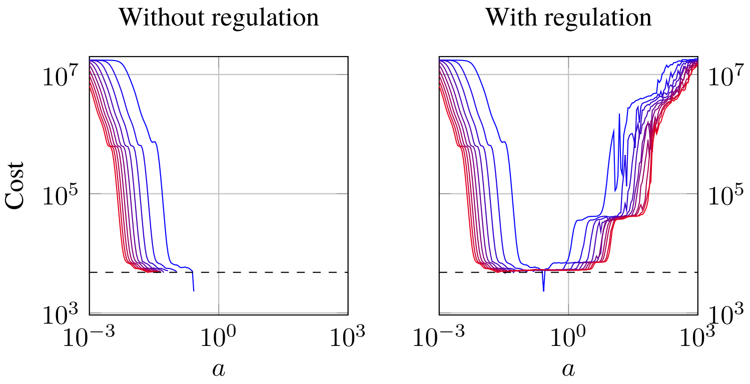


Figure 5.8 Minimal cost when solving P_f for different values of a and N with, and without regulation. To obtain the result in the right plot a robustness factor $\gamma = 4$ was used.

cantilever beams, excited by the vibratory motion of a shaker table. The system has a slightly dampened resonance at 110Hz. The goal is to find a controller so that the mass M_1 tracks reference signals up to 10Hz, while minimizing the displacement of M_2 at resonant frequency. In [Sznaier, 2000] the authors provide a 7th-order model with seven significant digits. We present the system, rounded to five significant digits in (5.1). The reason for lowering the number of significant digits is because of improved readability, and that we found no difference between using the original system and the rounded system in the context of control design.

$$\begin{aligned}
A &= \begin{bmatrix} -0.22 & 110.27 & -1.07 & -0.42 & -1.20 & 0 & -0.08 \\ -110.28 & -0.99 & 3.73 & 2.15 & 1.88 & -0.08 & 0.16 \\ -1.41 & -4.52 & -12.99 & 42.71 & -41.27 & -1.50 & -2.17 \\ -0.89 & -2.56 & -51.81 & -5.21 & -82.65 & 0.53 & 3.27 \\ 1.43 & 2.26 & 48.77 & 82.8 & -9.47 & 2.53 & -1.13 \\ 0.10 & 0.22 & 2.11 & 0.84 & -2.84 & -0.13 & -145.96 \\ -0.27 & -0.6 & -3.46 & -4.30 & 4.83 & 145.99 & -0.95 \end{bmatrix} \\
B &= \begin{bmatrix} 2.71 & 5.75 & 10.41 & 6.11 & -7.57 & -0.63 & 1.70 \end{bmatrix}^T \\
C &= \begin{bmatrix} 0.57 & -0.08 & 7.26 & -3.88 & -3.49 & 0.60 & 1.68 \\ 2.65 & -5.75 & 7.47 & 4.71 & 6.72 & -0.18 & 0.29 \end{bmatrix} \\
D &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{5.1}$$

The problem can be formulated as a constrained H_2 minimization problem:

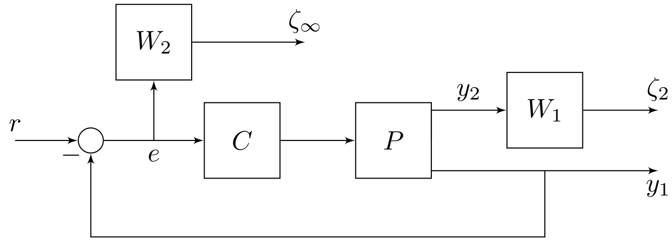


Figure 5.9 The system considered in Section 5.3

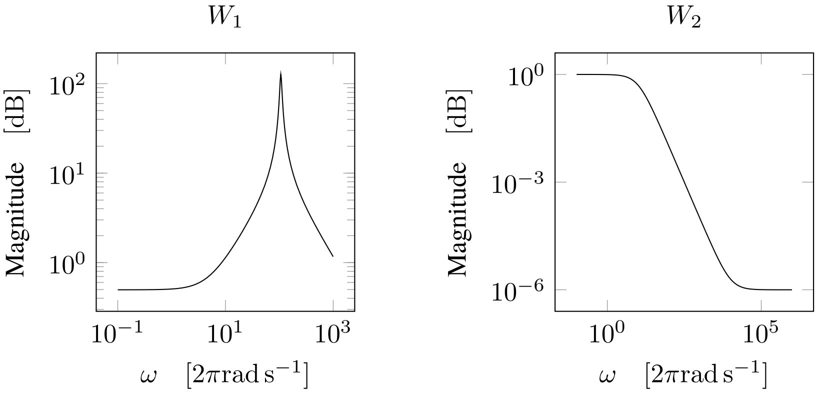


Figure 5.10 Bode magnitude plots of the notch filter W_1 and the low-pass filter W_2.

$$\begin{array}{ll} \text{minimize} & \|W_1 T_{y_2 r}\|_2 \\ \text{subject to} & \|W_2 T_{er}\|_\infty \leq \gamma \end{array}$$

where $e = r - y_1$ is the tracking error of M_1, and $W_1,\ W_2$ are weight functions given by:

$$W_1(s)\frac{1.1531s + 5.6566}{0.001s^2 + 0.0091s + 11.384}, \quad W_2(s) = \left(\frac{10^{-4}s + 1}{0.1s + 1}\right)^2 \tag{5.2}$$

where W_1 is a notch filter with a peak around 110Hz, and W_2 is a low pass filter. Bode plots of the filters are shown in Figure 5.10. The system is depicted in Figure 5.9

Previous results

The results reported in [Sznaier, 2000] are presented in Table 5.1

Table 5.1 Results reported by [Sznaier, 2000]

Type	Controller Order	$\|H_{\zeta_2 r}\|_2$	$\|H_{\zeta_\infty r}\|_\infty$
optimal $\mathcal{H}_\infty$	11	1534	0.47
mixed $\mathcal{H}_2/\mathcal{H}_\infty$ (LMI)	11	2168	0.51
mixed $\mathcal{H}_2/\mathcal{H}_\infty$	29	1254	0.51
mixed $\mathcal{H}_2/\mathcal{H}_\infty$	18	1271	0.52

Method

Let $P_1(s)$ be associated with y_1 and $P_2(s)$ with y_2. Then $H_{\zeta_2 w} = W_1 P_2 Q$ and $H_{\zeta_\infty w} = (W_2 - W_2 P_1 Q)$. Construct a vector ω of N_ω logarithmically spaced frequency points in $[10^{-4},\ 10^{6}]$. The problem then becomes

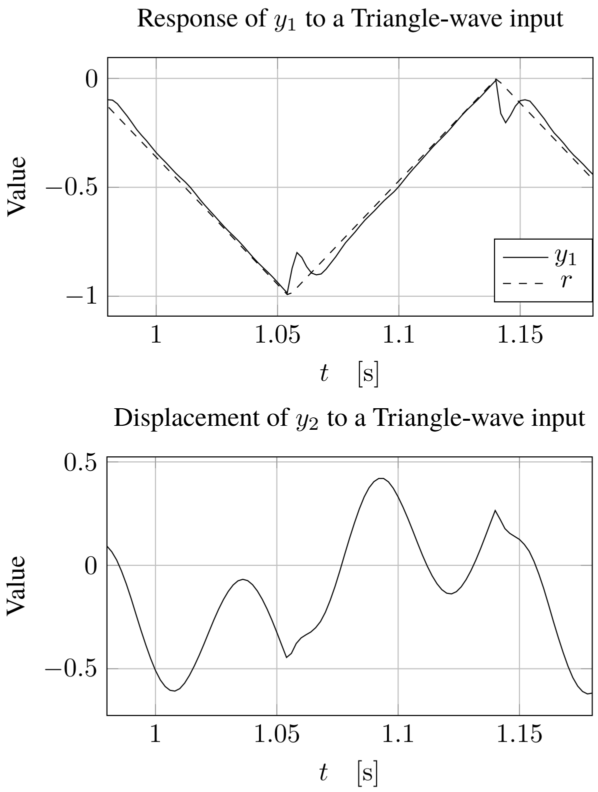
$$\begin{array}{ll} \text{minimize} & \|H_{\zeta_2 w}\|_2 \\ \text{subject to} & \|H_{\zeta_\infty w}\|_\infty \leq \gamma \end{array}$$

where the constraint can be approximated as

$$|H_{\zeta_\infty w}(i\omega_k)| \leq \gamma,\ \forall k = 1, \dots, N_\omega$$

Results

Setting $\gamma = 0.0051$ taking $N_\omega = 5000,\ N = 200$ and $a = 100$ results in the cost 1275. The output to a triangle wave, taking $f = 5.84$Hz is shown in Figure 5.11. The results differ from those reported in [Sznaier, 2000]. The difference lies in that the the bound γ is $1/100$th of what the authors claim. The cost is slightly higher, but the response to a triangle wave input seems similar.



Response of y_1 to a Triangle-wave input

Displacement of y_2 to a Triangle-wave input

Figure 5.11 Outputs to a triangle wave with $f = 5.84$Hz

6

Discussion

6.1 Convergence

It is clear from the previous chapter that the cutoff frequency of the Laguerre filters has a large influence on the rate of convergence. We also observed that right half-plane zeros forces the range to be very narrow. From all the problems presented in Section 5.1 it seems like the minimum cost is smooth, and possibly quasi-convex in the cutoff frequency. Quasi-convexity implies that gradient-descent methods can be used and thus the optimization scheme may be wrapped in an outer algorithm with the purpose to find optimal a. One approach to find a good candidate cutoff frequency is to arbitrarily select the cutoff frequency then perform balanced truncation of the Youla parameter to find a region where the rate of convergence is high.

6.2 Numerical issues

We found that `1lyap` and `4lyap` performed substantially better than `toep` and `lyapchol`, which seems to experience issues for larger values of a. On numerous occasions poor results can be attributed to the Hessian becoming indefinite. Since the Hessian should be positive semidefinite this is attributed to computational inaccuracies. Comparing Figures 3.4 and 3.5 to the results, it seems like the value of the smallest eigenvalue, which could be seen as a measure of how far a matrix is from being positive definite, is not a sufficient basis for evaluation of numerical stability, as the results of `toep` are unsatisfactory.

6.3 Applications in other areas

The method of computing inner products via solving a Sylvester equation, combined with the output-orthogonal realization of Laguerre filters can be used to project a system onto the Laguerre filters. This can be used to approximate any continuous stable linear system. Since bilinear transform of Laguerre filters results in powers of z^{-1} this approach can be used to compute finite impulse-response approximations.

6.4 Suggested further research

Based on the experiences gained and results observed we suggest the following research topics.

Weighted inner-product of impulse responses

We found out that it is of interest to use time-domain-weighted l_2 norms as cost functionals. If the weight function is a linear combination of piecewise exponential functions we may extend Theorem 3.4 to include a weight function in the following manner. Let

$$\begin{aligned} h_1(t) &= C_1 e^{A_1 t} B_1 \\ h_2(t) &= C_2 e^{A_2 t} B_2 \\ v(t) &= \begin{cases} ce^{at}, & t_0 \leq t < t_1 \\ 0, & \text{else} \end{cases} \end{aligned}$$

where $t_0 > 0$. Let the weighted inner-product for casual systems be defined by

$$\langle h_1, h_2 \rangle_v = \int_0^\infty h_1^*(t) h_2(t) v(t) dt$$

Inserting h_1 h_2 and v leads to

$$\begin{aligned} \langle h_1, h_2 \rangle_v &= \int_{t_0}^{t_1} B_1^T e^{A_1^T t} C_1^T C_2 e^{A_2 t} B_2 c e^{at} dt \\ &= B_1^T \underbrace{\int_{t_0}^{t_1} e^{A_1^T t} C_1^T C_2 e^{A_2 t} c e^{at} dt}_{W} B_2 \end{aligned}$$

Differentiating under the integral sign gives

$$A_1^T W + W A_2 + aW = e^{A_1^T t_1} C_1^T C_2 e^{A_2 t_1} c e^{a t_1} - e^{A_1^T t_0} C_1^T C_2 e^{A_2 t_0} c e^{a t_0}$$

To write the left hand side on standard Sylvester equation form take

$$\begin{aligned} \tilde{A}_1 &= A_1 + z_1 I \\ \tilde{A}_2 &= A_2 + z_2 I \end{aligned}$$

where

$$z_1 + z_2 = a$$

is taken such that the eigenvalues of $\tilde{A}_1$ and $\tilde{A}_2$ have negative real part.

$\mathcal{H}_\infty$-norm bounds for systems with time-delay

Consider a $\mathcal{H}_\infty$-norm constraint on the sensitivity function for a SISO system with time-delay. Let the sensitivity function be given by

$$S(s) = 1 - P_0(s)e^{-\tau s}Q(s)$$

We can derive a bound on the infinity-norm.

$$\begin{aligned} \|S\|_\infty^2 &= \sup_\omega \left(|1 - P_0(i\omega)e^{-\tau i\omega}Q(i\omega)|^2\right) \\ &= \sup_\omega \left(1 - 2\,\mathrm{Re}\big\{P_0(i\omega)e^{-\tau i\omega}Q(i\omega)\big\} + |P_0(i\omega)Q(i\omega)|^2\right) \\ &\leq \sup_\omega \left(1 + 2|P_0(i\omega)Q(i\omega)| + |P_0(i\omega)Q(i\omega)|^2\right) \end{aligned}$$

This bound could potentially be useful because the exponential part will make the sensitivity function oscillate quickly, possibly requiring an excessive amount of points in the frequency grid. This is illustrated in the following example.

Example 6.1—Sensitivity bound
Consider the system

$$\begin{aligned} P_1(s) &= \frac{e^{-s}}{1+s} \\ D(s) &= \frac{1}{1+10s} \\ N(s) &= 1 \end{aligned}$$

where the intensities of the process disturbance, and measurement noise are given by $W_{\text{process}} = 200$ and $W_{\text{sensor}} = 10^{-3}$, respectively and $\rho = 0.3$. The problem was solved exactly using $N = 50$ and $a = 100$ for the Ritz expansion. Bode's magnitude-plots for the sensitivity function and aforementioned bound are shown in Figure 6.1. Note that the bound is quite conservative for lower frequencies but is tight for the oscillation peaks for higher frequencies. The absolute value can be easily implemented using convex-optimization software when working with Youla-parametrized optimal-controller design. □

Other basis functions

This thesis primarily concerns the Laguerre bases. It would be possible to add notch filters, or Kautz filters (similar to Laguerre filters but allows for multiple, possibly complex conjugated poles). Other ideas are piecewise constant, piecewise linear functions, wavelets and Fourier series.

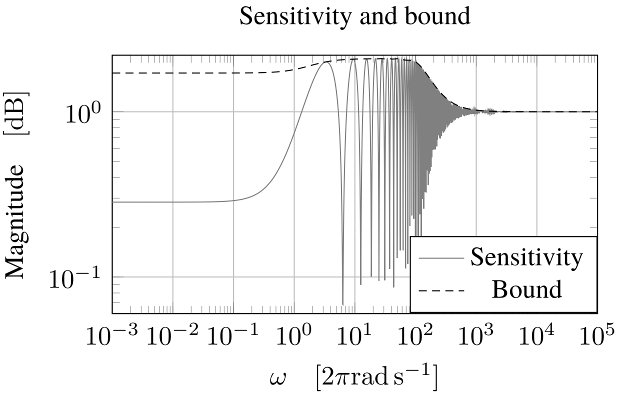


Figure 6.1 Magnitude of the sensitivity function and upper bound for Problem 1 solved exactly. $N = 50$ and $a = 100$.

6.5 Conclusion

We propose using `llyap` to compute finite-dimensional equivalents to quadratic cost-functionals on the subset of $\mathcal{H}_2$ spanned a truncated Ritz expansion. Furthermore we propose that reports of future research on Youla-parametrized optimal-controller design using Laguerre bases should include the strategy used to select the cutoff frequency of the filters. We believe that further improvements to the proposed algorithms may be possible, but are likely to lead to marginal improvements, and we suggest studying other basis functions, or combinations thereof instead. Even if other bases are used, `llyap` can be quickly implemented and executed to construct quadratic-program approximations of $\mathcal{H}_2$ cost functionals.

Bibliography

Boyd, S. and C. Barratt (1991). *Linear Controller Design: Limits of Performance*. Prentice-Hall.

Boyd, S. and L Vandenberge (2009). *Convex Optimization*. Cambridge University Press. ISBN: 978-0-521-83378-3.

Brawer, R. and M. Pirovino (1992). "The linear algebra of the Pascal matrix". *Linear Algebra and its Applications* **174**, pp. 13 –23. ISSN: 0024-3795. DOI: 10.1016/0024-3795(92)90038-C. URL: http://www.sciencedirect.com/science/article/pii/002437959290038C.

CPLEX (2018). *Convex optimization software*. URL: https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer (visited on 2018-05-29).

CVX (2018). *Matlab-based modeling system for convex optimization*. URL: http://cvxr.com/cvx/ (visited on 2018-05-29).

Glad, T and L Ljung (2016). *Reglerteori. Flervariabla och olinjära metoder*. Studentlitteratur. ISBN: 978-91-44-03003-6.

Gurobi (2018). *Convex optimization software*. URL: http://www.gurobi.com/ (visited on 2018-05-29).

Heuberger, P. S., P. M. Van den Hof, and B. Wahlberg (2005). *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer. ISBN: 1-85233-956-X.

Hägglund, T. and K. J. Åström (2004). "Revisiting the ziegler-nichols step response method for PID control". eng. Journal of Process Control **14**:6, pp. 635–650. ISSN: 1873-2771. DOI: 10.1016/j.jprocont.2004.01.002.

Ma, Y., J. Yu, and Y. Wang (2014). "Efficient recursive methods for partial fraction expansion of general rational functions". *Journal of Applied Mathematics* **2014**. Article ID: 895036.

MathWorks (2018). *MATLAB's optimization toolbox*. Convex Optimization Software. URL: https://se.mathworks.com/products/optimization.html (visited on 2018-05-29).

Megritski, A. (1994). “On the order of optimal controllers in the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control”. *Proceedings of the 33rd Conference on Decision and Control* **33**, pp. 3173 –3174.

MOSEK (2018). *Convex optimization software*. URL: `https://www.mosek.com/` (visited on 2018-05-29).

Sznaier, M. (1994). “An exact solution to general siso mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problems via convex optimization”. *IEEE Transactions on Automatic Control* **39**, pp. 2511 –2517.

Sznaier, M. (2000). “An exact solution to continuous-time mixed H_2/H_∞ control problems”. *IEEE Transactions on Automatic Control* **45**, pp. 2095 –2101.

Wahlén, E. (2014). *The matrix exponential*. URL: `http://www.ctr.maths.lu.se/media/MATC12/2014ht2014/exp_6.pdf`.

Wernrud, A. (2008). *QTool 0.1 - Reference Manual*. Department of Automatic Control, Lund University.

Youla, D. C., J. J. Bongiorno, and H. A. Jabr (1976a). “Modern Wiener-Hopf design of optimal controllers. Part 1: the single-input-output case”. *IEEE Transactions on Automatic Control* **AC-21**, pp. 3–13.

Youla, D. C., J. J. Bongiorno, and H. A. Jabr (1976b). “Modern Wiener-Hopf design of optimal controllers. Part 2: the multivariable case”. *IEEE Transactions on Automatic Control* **AC-21**, pp. 319 –338.

Zhou, K. and J. Doyle (1997). *Essentials of Robust Control*. Prentice Hall.

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER’S THESIS

Date of issue
June 2018

Document Number
TFRT-6061

Author(s)
Olle Kjellqvist

Supervisor
Olof Troeng, Dept. of Automatic Control, Lund University, Sweden
Pontus Giselsson, Dept. of Automatic Control, Lund University, Sweden
Anton Cervin, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Laguerre Bases for Youla-Parametrized Optimal-Controller Design:
Numerical Issues and Solutions

Abstract

This thesis concerns the evaluation of cost functionals on H2 when designing optimal controllers using finite Youla parameterizations and convex optimization. We propose to compute inner products of stable, strictly proper systems via solving Sylvester equations. The properties of different state space realizations of Laguerre filters, when performing Ritz expansions of the optimal controller are discussed, and a closed form expression of the output orthogonal realization is presented. An algorithm to exploit Toeplitz substructure when solving Lyapunov equations is discussed, and a method to extend SISO results to MIMO systems using the vectorization operator is proposed. Finally the methods are evaluated on example systems from the industry, where it is shown that properly selecting the cutoff frequency of the filters is an important problem that should be discussed when Laguerre bases are used to parametrize the optimal controller.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-67

Recipient’s notes

Security classification

http://www.control.lth.se/publications/