Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Numerical Pitfalls in Q-Design

Olle Kjellqvist * **Olof Troeng** *

* *Department of Automatic Control, Lund University, Sweden*
E-mail: {olle.kjellqvist,oloft}@control.lth.se

.

Abstract: Q-design is a powerful method for designing approximately optimal LTI controllers and assessing the achievable control performance. Unfortunately, numerical issues are often encountered in Q-design which limits its applicability. This paper warns about two numerical pitfalls in Q-design when using $\mathcal{H}_2$ costs and Laguerre-type basis functions.

1. INTRODUCTION

Many control design problems can be formulated as optimization of transfer functions between external inputs (measurement noise, disturbances, reference signals) to external outputs (controlled variables and control errors) (Zhou et al., 1996). The objective and constraints on the transfer functions are often convex, in which case an approximately optimal LTI controller can be found by the so-called Q-design (Boyd and Barratt, 1991). Q-design is a useful tool for exploring the limits of control performance (Boyd and Barratt, 1991), and for benchmarking of low-order controllers (e.g., PID) against the optimal LTI controller (Garpinger, 2009; Larsson and Hägglund, 2011). Q-design can also be used as a basis for control design (Du et al., 2006; Scherer, 1995), where the latter contributed details on the convergence of finite-dimensional approximations.

Although the basic principles of Q-design are straightforward and well known in the control community, it has seen little use relative to its utility. We believe that a main reason is that the resulting convex programs tend to be numerically ill-conditioned, which makes solvers fail from numerical issues. To avoid ill-conditioning it is necessary to select the basis functions carefully and to keep their number small, which limits the usability of Q-design.

In this contribution, we consider numerical issues that arise in connection with quadratic terms and two popular basis expansions for the Q parameter. First, we need to introduce some background to explain our contributions adequately.

Q-design

The Q-parameterization (Youla et al., 1976; Zhou et al., 1996; Boyd and Barratt, 1991) parametrizes all internally stabilizing LTI controllers in a parameter $Q \in \mathcal{H}_\infty$, so that the closed-loop transfer function from external inputs w to external outputs z becomes affine in Q, i.e.,

$$H_{zw} = T_1 + T_2 Q T_3, \tag{1}$$

with $T_1, T_2, T_3 \in \mathcal{H}_\infty$. This makes many closed-loop properties convex in Q, e.g., $\mathcal{H}_2$-norms, $\mathcal{H}_\infty$-norms, and overshoot (Boyd and Barratt, 1991).

Q-design (Boyd and Barratt, 1991) amounts to approximating Q with a linear combination of a finite-dimensional basis $\{q_k\}_{k=1}^N$. We will assume that Q is single-input single-output (SISO), since this is sufficient to illustrate our main points. In this case, Q can be expanded as

$$Q = \sum_{k=1}^{N} \beta_k q_k = G_q \beta, \quad q_k \in \mathcal{H}_\infty, \tag{2}$$

where the coefficient vector $\beta = [\beta_1 \cdots \beta_n]^\mathsf{T} \in \mathbb{R}^N$, and

$$G_q = [q_1 \ q_2 \ \cdots \ q_n] \tag{3}$$

is a $1 \times N$ transfer function matrix. If the objective and constraints on H_{zw} are convex then standard optimization software can find the optimal coefficients $\beta^\star$, from which the corresponding controller can be recovered (Boyd and Barratt, 1991).

Considered Bases

We will consider two closely related families of bases. What we call *repeated-pole expansions* (Boyd and Barratt, 1991, (15.10))(Hespanha, 2009),

$$q_k(s) = \left(\frac{a}{s+a}\right)^k, \qquad k = 1, 2, \ldots \tag{4}$$

and *Laguerre bases* (Heuberger et al., 2005)

$$q_k(s) = \frac{\sqrt{2a}}{s+a} \cdot \left(\frac{a-s}{a+s}\right)^{k-1}, \qquad k = 1, 2, \ldots \tag{5}$$

where a is real and positive. It is clear that their truncations to the first N terms have the same span. The constant term, $k = 0$, is omitted to ensure finite $\mathcal{H}_2$ cost.

Quadratic Costs

We will consider 2-norms of transfer function matrices between *subsets* of the inputs z and outputs w, i.e., of a submatrix of (1). Such terms can be written

$$\|G_1 + G_2 Q\|_2 = \|G_1 + G_2 G_q \beta\|_2, \tag{6}$$

where G_1 and G_2 single-input multiple-output transfer functions [1]. To include terms of the form (6) in a convex

⋆ This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 834142 (ScalableControl). The authors are members of the ELLIIT Strategic Research Area at Lund University.

[1] A submatrix of H_{zw} can be written $\widetilde{H}_{zw} = \widetilde{T}_1 + \widetilde{T}_2 Q \widetilde{T}_3$. Let $G_1 := \mathrm{vec}(\widetilde{T}_1)$ and $G_2 := \mathrm{vec}(\widetilde{T}_2 \widetilde{T}_3)$ where vec denotes vectorization (stacking the matrix columns into a vector). Then, since Q is SISO, $\|\widetilde{H}_{zw}\|_2 = \|G_1 + G_2 Q\|_2$. See (Kjellqvist, 2018) for the MIMO case.

Copyright lies with the authors

program they need to be written as quadratic forms in the coefficients [2] $\beta \in \mathbb{R}^N$,

$$\|G_1 + G_2 G_q \beta\|_2^2 = \beta^\mathsf{T} M \beta + 2m^\mathsf{T}\beta + \|G_1\|_2^2, \qquad (7)$$

where $M \in \mathbb{R}^{N\times N}$, and $m \in \mathbb{R}^N$. The computational details of obtaining M and m will be discussed in Section 2.

Contributions

1. We show that the repeated-pole expansions (4) typically give poorly conditioned optimization problems and that the Laguerre bases (5) are a better option. This is an important observation since the repeated-pole expansion is prevalent in existing Q-design literature (Boyd and Barratt, 1991, Ch. 15), (Hespanha, 2009). A rare use of Laguerre bases is seen in (Ferreres, 2014).

2. We show that state-space-based computation (Sec. 2) of M and m in (7) is prone to numerical problems if the realization of G_q is not carefully selected. We give an example of a natural-looking realization of the Laguerre bases that leads to horribly conditioned matrix computations. We suggest a realization of the Laguerre bases that avoids these problems.

3. We show by an example that there is a trade-off in the choice of a in (5) between the conditioning of M, and the ability to approximate the optimal, infinite-dimensional Q.

2. REDUCTION OF 2-NORMS TO FINITE-DIMENSIONAL QUADRATIC FORMS

To find the matrix M and vector m in (7) we write

$$\begin{aligned}\|G_1 + G_2 Q\|_2^2 &= \|G_1 + G_2 G_q \beta\|_2^2 \\ &= \|G_1\|_2^2 + 2\langle G_1, G_2 G_q \beta\rangle + \|G_2 G_q \beta\|_2^2.\end{aligned}$$

Introducing state-space realizations

$$\left[\begin{array}{c|c} A_i & B_i \\ \hline C_i & 0 \end{array}\right], \quad i = 1, 2,$$

where $D_i = 0$ follows from requiring the norm to be finite. For G_1, and $G_2 G_q$ we have

$$2\langle G_1, G_2 G_q \beta\rangle = \underbrace{2\int_0^\infty B_1^\mathsf{T} e^{A_1^\mathsf{T} t} C_1^\mathsf{T} C_2 e^{A_2 t} B_2 dt}_{=m^\mathsf{T}} \beta,$$

$$\|G_2 G_q \beta\|_2^2 = \beta^\mathsf{T} \underbrace{\int_0^\infty B_2^\mathsf{T} e^{A_2^\mathsf{T} t} C_2^\mathsf{T} C_2 e^{A_2 t} B_2 dt}_{=M} \beta.$$

We can now compute $m^\mathsf{T} = 2B_1^\mathsf{T} X B_2$, where $X = \int_0^\infty B_1^\mathsf{T} e^{A_1^\mathsf{T} t} C_1^\mathsf{T} C_2 e^{A_2 t} dt$ solves the Sylvester equation

$$A_1^\mathsf{T} X + X A_2 + C_1^\mathsf{T} C_2 = 0. \qquad (8)$$

Similarly, $M = B_2^\mathsf{T} Y B_2$, where the observability Gramian Y solves

$$A_2^\mathsf{T} Y + Y A_2 + C_2^\mathsf{T} C_2 = 0. \qquad (9)$$

Remark: In Matlab it is typically preferable to use the command `lyap` rather than `sylv` to solve (8).

Remark: To include a term (7) in a second-order cone program requires a Cholesky factor $M^{1/2}$ of M. The elegant method in (Hammarling, 1982) allows $M^{1/2}$ to be obtained directly from (9).

[2] For complex-coefficient systems (Troeng et al., 2017), let $\beta \in \mathbb{C}^N$, replace all T's by $*$'s, and change the term $2m^\mathsf{T}\beta$ in (7) to $2\,\mathrm{Re}\{m^*\beta\}$.

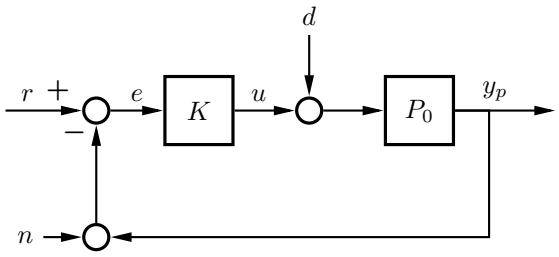


Figure 1. Standard feedback interconnection.

3. NUMERICAL ISSUES

3.1 Example System

To exemplify the numerical issues that we discuss in this section, we consider the standard feedback interconnection in Fig. 1, with a stable SISO plant P_0. A parametrization of internally stabilizing controllers is then given by $K = Q/(1 - P_0 Q)$, $Q \in \mathcal{H}_\infty$ (Zhou et al., 1996, Thm 12.7). With this parametrization the transfer function from measurement noise n to control action u becomes

$$H_{un} = K/(1 + P_0 K) = Q.$$

In control design, it is often important to limit the amplification of measurement noise to the control signal, which corresponds to a bound on $\|H_{un}\|_2 = \|Q\|_2$. This motivates that a basis expansion for Q-design needs to give a numerically well-conditioned Hessian M in

$$\|Q\|_2^2 = \|G_q \beta\|_2^2 = \beta^\mathsf{T} M \beta. \qquad (10)$$

The simplistic case (10) is considered in the following sections to allow easy analytic computations that illustrate our main points. Our numerical example in Sec. 4 show that these issues also arise in practical Q-design.

In Sec. 3.2 we consider how the choice of basis affects the conditioning of M, and in Sec. 3.3 we discuss how a poor realization of G_q gives numerical issues when computing M according to Sec.2.

3.2 How the Q basis affects conditioning of the Hessian M

A state-space realization of G_q for the repeated-pole expansion (4) is given by

$$A_q = \begin{bmatrix} -a & 1 & & \\ & -a & \ddots & \\ & & \ddots & 1 \\ & & & -a \end{bmatrix}, \qquad (11a)$$

$$B_q = \begin{bmatrix} a & & & \\ & a^2 & & \\ & & \ddots & \\ & & & a^N \end{bmatrix}, \qquad (11b)$$

$$C_q = [1\ 0\ \cdots\ 0]. \qquad (11c)$$

For this realization it can be shown (App. A) that the Y in (9) takes the form

$$Y = \frac{1}{2a}\,\mathrm{diag}(1, 1/2a, (1/2a)^2, \ldots) \cdot P_S \cdot \mathrm{diag}(1,\ 1/2a,\ (1/2a)^2, \ldots), \qquad (12)$$

where P_S is the symmetric $N \times N$ Pascal matrix (defined in (A.2)). It follows that the Hessian M is given by

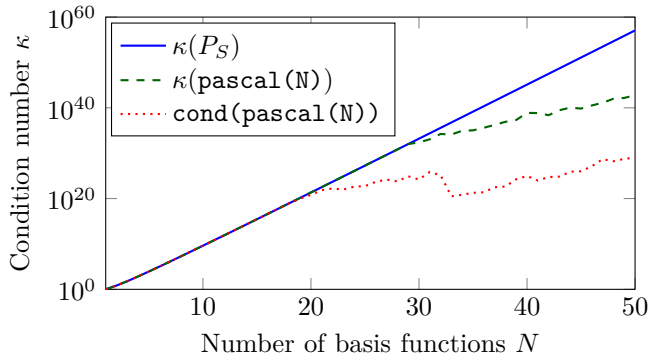


Figure 2. Condition number of symmetric Pascal matrices P_S as a function of their dimension N. The blue, solid line shows the true condition number ($\kappa(P_S) \sim 16^N/(N\pi)$ (Higham, 1996, Sec. 26.4)). The green, dashed line shows the condition number of MatLab's slightly inaccurate `pascal(N)` computed with variable precision arithmetic (`vpa`, 150 digits accuracy). The red, dotted line shows the condition number of `pascal(N)` computed with double-precision arithmetic. Note that the true condition number of P_S turns out to be significantly worse than what can be inferred with double-precision arithmetic.

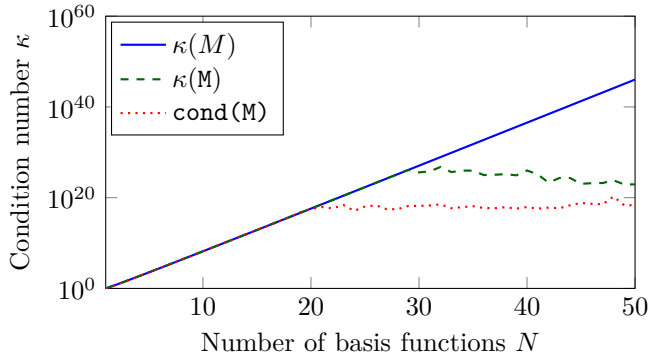


Figure 3. Condition number of the Hessian M in (13) computed with the same three approaches as in Fig. 2.

$$M = \frac{a}{2}\,\mathrm{diag}(1,\, 1/2,\, 1/2^2,\, \ldots)\cdot P_S \cdot \mathrm{diag}(1,\, 1/2,\, 1/2^2,\, \ldots). \tag{13}$$

Unfortunately, Pascal matrices are poorly conditioned (see (Alonso et al., 2013; Higham, 1996), and Fig. 2). The same goes for M as seen in Fig 3. The poor conditioning of (13) may give optimization software numerical problems for even a modest number N of basis functions. Note that the poor conditioning of M is independent of the realization of G_q. For the Laguerre basis (which is orthonormal), we have that $M = I$, and hence the condition number $\kappa(M) = 1$.

3.3 How the realization of G_q can give problems when computing the Hessian M

A natural realization of the transfer function matrix G_q in (3), for the Laguerre basis (5), would be to use the same A_q and C_q as in (11), and [3]

$$B_q = \sqrt{2a}\,\mathrm{diag}\big(1, 2a, \ldots, (2a)^{N-1}\big)\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & \cdots \\ 0 & 1 & -2 & 3 & \cdots \\ 0 & 0 & 1 & -3 & \cdots \\ 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{P_U^{-1}}. \tag{14}$$

The Gramian Y for computing $M = B_q^{\mathsf{T}} Y B_q$ is the same as in (12). Thus, the condition numbers of both Y and B_q grow exponentially with respect to N, preventing reliable computations of M with the approach in Sec. 2.

Note that in the specific case that we consider here ($G_2 = I$), we already know that $M = I$ since the basis is orthonormal. As we see in the next section, the same numerical issues also arise in the general case $G_2 \neq 1$.

One approach to alleviate the numerical issues is to insist that the realization of G_q has the following property.

Definition 1. A state-space realization is *output-orthogonal* if the observability Gramian is the identity matrix.

An output-orthogonal realization (i.e., $Y = I$) of the Laguerre basis is seen to be given by

$$A_q = \begin{bmatrix} -a & 2a & -2a & \cdots & (-1)^N 2a \\ & -a & 2a & \ddots & \vdots \\ & & -a & \ddots & -2a \\ & & & \ddots & 2a \\ & & & & -a \end{bmatrix}, \tag{15}$$

$$C_q = \sqrt{2a}\,[1\ -1\ 1\ -1\ \cdots],\ B_q = I.$$

See App. B for a full derivation. It should be noted that neither A_q nor C_q has elements that increase in magnitude with increasing N, and that A_q is on triangular form allowing stable computations of quadratic forms (7).

4. DESIGN EXAMPLE

We now illustrate the two numerical issues from the previous section with a simple design example. We also consider how the impact of the parameter a in (4) and (5) affects the approximation performance.

Example Problem from (Boyd and Barratt, 1991, Sec 2.4)

Consider the feedback interconnection in Fig. 1 with

$$P_0(s) = \frac{1}{s^2}\cdot\left(\frac{10-s}{10+s}\right). \tag{16}$$

The process disturbance d, and the measurement noise n are stationary white processes with intensities $W_d^2 = 0.04^2$ and $W_n^2 = 0.01^2$, respectively. The goal is to find the optimal LTI controller with respect to the design specification

$$\begin{aligned} &\text{minimize} && J = \|u\|_2 \\ &\text{subject to} && \|y_p\|_2 \leq 0.1. \end{aligned} \tag{17}$$

This corresponds to minimizing the rms control signal activity, subject to a constraint on the control error.

The optimum of (17) is $J^\star = 0.0397$ (Boyd and Barratt, 1991, p. 355).

[3] P_U^{-1} in the expression below is the inverted upper-triangular Pascal matrix (Brawer and Pirovino, 1992), for which $P_S = P_U^{\mathsf{T}} P_U$; thus $\kappa(P_S) = \sqrt{\kappa(P_U)}$.

Finding Approximately Optimal Controller with Q-design

The plant (16) is not asymptotically stable, so we use the following nominal controller (Boyd and Barratt, 1991, Sec 2.4)

$$K_0(s) = \frac{44.14s^2 + 107.3s + 39}{s^3 + 10s^2 + 55.25s + 78.14} \tag{18}$$

to stabilize the system, allowing the relevant closed loop transfer functions to be written as (1) with $w = [d\ n]^\mathsf{T}$ to $z = [u\ y_p]^\mathsf{T}$.

The quantities of interest in (17) can be expressed using the 2-norms of the closed-loop transfer functions,

$$\|u\|_2^2 = \|H_{ud}\|_2^2 W_d^2 + \|H_{un}\|_2^2 W_n^2, \tag{19a}$$

$$\|y_p\|_2^2 = \|H_{y_p d}\|_2^2 W_d^2 + \|H_{y_p n}\|_2^2 W_n^2. \tag{19b}$$

Applying the method in Sec. 2 to each 2-norm $\|H_{\mu\nu}\|_2^2$ gives us $M_{\mu\nu}$, $m_{\mu\nu}$, and $c_{\mu\nu}$, and lets us re-write (17) as

$$\begin{aligned} &\text{minimize} && J = \beta^\mathsf{T} M_u \beta/2 + m_u \beta + c_u \\ &\text{subject to} && \beta^\mathsf{T} M_y \beta/2 + m_y \beta + c_y \le 0.1^2 \end{aligned} \tag{20}$$

where $M_u = M_{ud} W_d^2 + M_{un} W_n^2$, etc. Now, (20) can be solved by standard optimization software.

Results

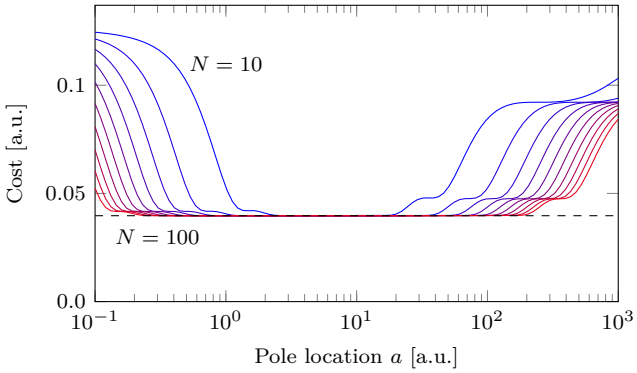
To numerically solve (20) we used `cvx` with Gurobi as the solver. Figs. 4a and 4b show the resulting costs for the Laguerre basis expansion (on output-orthogonal form) and the repeated-pole expansion, respectively. For given N and a the two expansions have the same span, which should imply that the figures are identical—however, this is clearly not the case. In Fig. 4b it is seen that repeated-pole expansion gives sub-optimal solutions for small a and makes Gurobi fail for large a. The problems for large as can be traced back to the poor numerical conditioning of M_u illustrated in Fig. 5 (consistent with Fig. 3). The suboptimal solutions for small a might be better explained by the matrix multiplication $B^\mathsf{T} Y B$, where B, given by (11b), has increasing powers of a on the diagonal. Picking $a \ll 1$ results in elements that quickly approach zero, effectively removing the influence of higher-order basis functions. Comparing the poor condition numbers of M_u when using the repeated-pole expansion with the better ones when using Laguerre basis functions in Fig. 5, we find—just like in the special case of Sec. 3.2—that Laguerre bases give superiorly conditioned quadratic programs.

Another interesting observation can be made by comparing Figs. 4a and 4b; the region of the lowest cost and the region of the best numerical conditioning are not the same. This means that there is a trade-off in the parameter a in (5) between the numerical conditioning of M and the ability to approximate the optimal Q. The impact of this for the simple problem in this section is small, but this may not be the case for larger, difficult-to-solve problems.

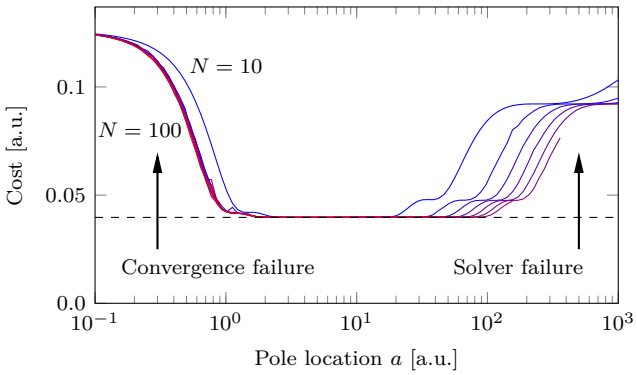
Computing M for Laguerre Bases with Jordan Realization

In Sec. 3.3 we saw that the choice of realization of G_q was crucial to get reasonable conditioning in the computation of Hessians M. As an illustration, we consider the computation (in double precision) of M_u in (20) for the Laguerre basis (5) with $a = 4$ and $N = 20$.

Using the Jordan realization ((11a), (14), (11c)) for G_q, we get an M_u with $\lambda_{\min}(M_u) = -7 \times 10^{-4}$ and



(a) Optimal cost of (20) when using Laguerre bases.



(b) Optimal cost of (20) when using repeated-pole expansions.

Figure 4. Optimal costs of (20) (solid lines) and true optimal cost $J^\star$ (dashed) plotted against the Pole location a. The number of basis functions ranges from $N = 10$ (blue) to $N = 100$ (red) in increments of 10. We see that the rate of convergence is highly dependent on a and that for $a < 1$ increasing N improves optimal cost when using Laguerre basis functions, but not repeated-pole expansions. For $a > 100$ and $N > 60$ the solver fails to find a solution when using a repeated-pole expansion, whereas it is possible to increase N further if the Laguerre basis functions are used.

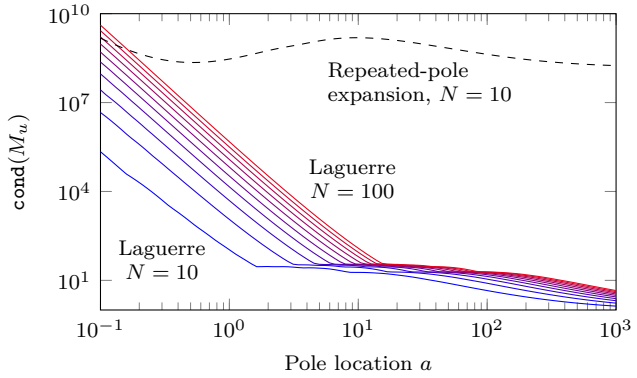


Figure 5. The condition number (double precision) of M_u when using Laguerre basis expansions with $N = 10$ (blue) to $N = 100$ (red), and repeated-pole expansions with $N = 10$ (dashed) against the pole location a. The condition number of M_u for the repeated pole expansions was worse than 10^{16} for $N > 20$.

$\lambda_{\max}(M_u) = 3 \times 10^{-3}$. Clearly, this M_u is far from positive semi-definite, indicating severe numerical errors in its computation. The output orthogonal realization (15) gives a matrix M_u with $\lambda_{\min}(M_u) = 9 \times 10^{-5}$ and $\lambda_{\max}(M_u) = 3 \times 10^{-3}$, that is close to the correct matrix.

5. CONCLUSIONS

We have seen that although the repeated-pole expansion (4) has received much attention in the Q-design literature, it typically gives poorly conditioned optimization problems. Clearly, the Laguerre basis seems to be numerically preferable. We have also illustrated that the choice of state-space realization is crucial when computing quadratic forms for the objective and constraints.

We hope that this article has shed light on some common numerical issues in Q-design and that a better understanding of these issues will allow wider adoption of Q-design throughout the control community.

Appendix A. ANALYTIC EXPRESSION FOR (12)

With A and C given by (11a) and (11c) the elements of the solution Y to the Lyapunov equation (9) satisfies the two-dimensional recursion equation

$$\begin{aligned} Y_{j,k} &= \frac{Y_{j-1,k} + Y_{j,k-1}}{2a}, \\ Y_{1,1} &= \frac{1}{2a}. \end{aligned} \tag{A.1}$$

The solution to (A.1) is given by

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

Introducing the symmetric Pascal matrix

$$P_S := \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1 & 2 & 3 & 4 & \cdots \\ 1 & 3 & 6 & 10 & \cdots \\ 1 & 4 & 10 & 20 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{A.2}$$

we can write

$$Y = \frac{1}{2a} \operatorname{diag}(1, 1/2a, (1/2a)^2, \ldots) \cdot P_S \cdot \operatorname{diag}(1,\ 1/2a,\ (1/2a)^2, \ldots).$$

Appendix B. DERIVATION OF (15)

The Laguerre bases (5) are given by the recursion

$$\begin{aligned} q_1(s) &= \frac{\sqrt{2a}}{a+s} \\ q_k(s) &= \left(\frac{a-s}{a+s}\right) q_{k-1}(s). \end{aligned}$$

Starting at $q_k(s)$ we can get a closed form solution by working our way backwards,

$$\begin{aligned} sq_k(s) &= -aq_k(s) + (a-s)q_{k-1}(s) \\ &= -aq_k(s) + aq_{k-1}(s) \\ &\quad - (-aq_{k-1}(s) + (a-s)q_{k-2}(s)) \end{aligned}$$

Expanding each $sq_{k-l}(s)$ results in

$$= -aq_k(s) + 2a \sum_{i=1}^{k-1} (-1)^{i+1} q_{k-i}(s) + (-1)^{k-1}\sqrt{2a}.$$

By taking $x_k(s) = q_k(s)u(s)$ as the internal states the realization of the differential equation $y(s) = \sum_{k=1}^{N} \beta_k q_k(s) u(s)$ becomes

$$A = \begin{bmatrix} -a & 0 & 0 & \\ 2a & -a & 0 & \\ -2a & 2a & -a & \\ & \ddots & \ddots & \ddots \end{bmatrix}, B = \sqrt{2a} \begin{bmatrix} 1 \\ -1 \\ 1 \\ \vdots \end{bmatrix}, C = \beta^{\mathsf{T}}$$

where β is a column vector containing all β_k. To better fit the context of the article we use the transposed version $\hat{A} = A^{\mathsf{T}}$, $\hat{B} = C^{\mathsf{T}}$ and $\hat{C} = B^{\mathsf{T}}$. Factoring out β from $\hat{B} = \beta I$ gives the realization (15).

REFERENCES

Alonso, P., Delgado, J., Gallego, R., and Peña, J.M. (2013). Conditioning and accurate computations with pascal matrices. *J. Comput. Appl. Math.*, 252, 21–26.

Boyd, S. and Barratt, C. (1991). *Linear Controller Design: Limits of Performance*. Prentice-Hall. URL `http://stanford.edu/~boyd/lcdbook/`.

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

Du, C., Xie, L., Guo, G., Zhang, J., Li, Q., Hredzak, B., and Teoh, J.N. (2006). A generalized KYP lemma based control design and application for 425 kTPI servo track writing. In *Proc. Am. Control Conf.* doi:10.1109/ACC.2006.1656397.

Ferreres, G. (2014). Convex design of a youla parameter for lti and lft plant models. Technical report, The French Aerospace Lab.

Garpinger, O. (2009). *Design of Robust PID Controllers with Constrained Control Signal Activity*. Licentiate thesis, Lund University, Sweden.

Hammarling, S.J. (1982). Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. of Numerical Analysis*, 2(3), 303–323.

Hespanha, J.P. (2009). *Linear Systems Theory*. Princeton University Press.

Heuberger, P.S., van den Hof, P.M., and Wahlberg, B. (2005). *Modelling and Identification with Rational Orthogonal Basis Functions*. Springer.

Higham, N.J. (1996). *Accuracy and Stability of Numerical Algorithms*. SIAM.

Kjellqvist, O. (2018). *Laguerre Bases for Youla-Parametrized Optimal-Controller Design: Numerical Issues and Solutions*. Master's thesis, Lund University, Sweden.

Larsson, P.O. and Hägglund, T. (2011). Control signal constraints and filter order selection for PI and PID controllers. In *Proc. Am. Control Conf.*

Scherer, C.W. (1995). Multiobjective $\mathcal{H}_2/\mathcal{H}_\infty$ control. *IEEE Trans. Autom. Control*, 40, 1054–1062.

Troeng, O., Bernhardsson, B., and Rivetta, C. (2017). Complex-coefficient systems in control. In *Proc. Am. Control Conf.*

Youla, D., Jabr, H., and Bongiorno, J. (1976). Modern Wiener-Hopf design of optimal controllers–part II: The multivariable case. *IEEE Trans. Autom. Control*, 21(3), 319–338.

Zhou, K., Doyle, J.C., and Glover, K. (1996). *Robust and Optimal Control*. Prentice Hall, New Jersey.