Parametric robust H_2 control

Laleh Ravanbod-Hosseini * Dominikus Noll * Pierre Apkarian †

Abstract

H_2-control with structured controllers is discussed, and a way to enhance the robustness of the design with respect to real uncertain parameters system is proposed.

Keywords: Structured H_2 control, parametric robustness.

1 Introduction

It is well-known that LQG or H_2-controllers often lack robustness with respect to plant uncertainty. Here we consider the situation when the plant has uncertain real parameters. A theoretical tool to model parametric uncertainty is the structured singular value μ_Δ introduced by Doyle [4], but its computation is known to be NP-complete, [2, 3, 12], which makes it unfit for use within an optimization procedure, where functions are called repeatedly. It is therefore mandatory to use approximations of μ_Δ or other heuristic criteria, which are suited in constrained optimization programs. Here we propose a new method which robustifies a given H_2-performance index $\mathcal{P}(G,K) = \|T_{w\to z}(G,K)\|_2^2$ by minimizing variations $\nabla_{\mathbf{p}}\mathcal{P}(G(\mathbf{p}),K)$ with respect uncertain parameters $\mathbf{p}$ in the system.

A classical way to address the lack of robustness in LQG is the well-known LQG/LTR procedure [14], which gains robustness by trading it against a loss of performance. We compare our new approach to LQG/LTR.

2 Preparation

2.1 Structured controllers

A controller in state-space form

$$(1) \qquad K : \begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \left[\begin{array}{c|c} A_K & B_K \\ \hline C_K & D_K \end{array}\right] \begin{bmatrix} x_K \\ y \end{bmatrix}$$

is called *structured* if the matrices A_K, B_K, C_K, D_K depend smoothly on a design parameter $\mathbf{x}$,

$$A_K = A_K(\mathbf{x}), B_K = B_K(\mathbf{x}), C_K = C_K(\mathbf{x}), D_K = D_K(\mathbf{x}),$$

varying in some parameter space $\mathbb{R}^n$, or in a constrained subset of $\mathbb{R}^n$. Here $n = \dim(\mathbf{x})$ is typically smaller than $\dim(K) = n_K^2 + m_2 n_K + p_2 n_K + m_2 p_2$, where m_2 is the number of inputs, p_2 the number of outputs, n_K the order of K. We also expect $n_K \ll n_x$, even though this is not formally imposed. Full order controllers satisfy $n_K = n_x$ and $\dim(\mathbf{x}) = \dim(K)$ and are referred to as *unstructured*.

Typical examples of controller structures are observer-based controllers

$$(2) \qquad K_{\text{obs}}(\mathbf{x}) = \left[\begin{array}{c|c} A - B_2K_c - K_fC_2 & K_f \\ \hline -K_c & 0 \end{array}\right],$$

where $\mathbf{x} = (\text{vec}(K_c), \text{vec}(K_f)) \in \mathbb{R}^{n_x m_2 + n_x p_2}$. Other practically useful controller structures include PID, decentralized and reduced-order controllers, or even entire synthesis structures combining controllers and filters.

2.2 Structured H_2 problem

Given a transfer matrix in standard form

$$(3) \qquad G(s) = \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & 0 & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right],$$

the structured H_2 synthesis problem is the following optimization program

$$(4) \qquad \begin{array}{ll} \text{minimize} & \mathcal{P}(\mathbf{x}) = \|T_{w\to z}(G, K(\mathbf{x}))\|_2^2 \\ \text{subject to} & K(\mathbf{x}) \text{ internally stabilizing}, \mathbf{x} \in \mathbb{R}^n \end{array}$$

In contrast with the standard H_2 control problem [15, 14.2], where the observer-based structure (2) arises by itself, (4) imposes the controller structure $K(\mathbf{x})$ as a constraint. In consequence, (4) is generally non-convex and more difficult to solve than the standard H_2 problem, and we accept locally optimal solutions. We refer to $\mathcal{P}(\mathbf{x})$ as the nominal performance, or simply as the performance. The solution $\mathbf{x}^{\text{nom}}$ of (4) is called the nominal design, $K(\mathbf{x}^{\text{nom}})$ the nominal controller, and $p^{\text{nom}} = \mathcal{P}(\mathbf{x}^{\text{nom}})$ the nominal performance.

2.3 Augmented system

In order to alleviate the notational burden of the formulas to come, we shall employ a standard trick to render the feedback controller (1) static. The plant G is artificially augmented by

$$A^{\text{aug}} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, B_2^{\text{aug}} = \begin{bmatrix} B_2 \\ 0 \end{bmatrix}, B^{\text{aug}} = \begin{bmatrix} 0 & B \\ I_k & 0 \end{bmatrix},$$

*Université Paul Sabatier, Institut de Mathématiques, Toulouse, France {L.Hosseini, D.Noll}@math-univ.fr

†ONERA, Department of Systems Control and Flight Dynamics, Toulouse, France {P.Apkarian}@onera.fr

$$C_2^{\text{aug}} = \begin{bmatrix} C_2 & 0 \end{bmatrix}, C^{\text{aug}} = \begin{bmatrix} 0 & I_k \\ C & 0 \end{bmatrix},$$
$$D_{12}^{\text{aug}} = \begin{bmatrix} 0 & D_{12} \end{bmatrix}, D_{21}^{\text{aug}} = \begin{bmatrix} 0 & D_{21} \end{bmatrix}.$$

Switching back from G^{aug} to G for notational convenience, we may without loss compute controllers $K(\mathbf{x})$ which are static, and at the same time structured.

3 Trade-off via mixed synthesis

The situation we are concerned with is when the open-loop system $G(\mathbf{p})$ contains uncertain parameters $\mathbf{p}$. Assuming that the nominal parameter values are $\mathbf{p}_0$, so that $G = G(\mathbf{p}_0)$, we wish to synthesize $K(\mathbf{x}^{\text{rob}})$ in such a way that it still performs well if $\mathbf{p}$ differs significantly from $\mathbf{p}_0$. A general heuristic strategy is to introduce a robustness function $\mathcal{R}(\mathbf{p}, \mathbf{x})$ which when minimized over $\mathbf{x}$ for fixed $\mathbf{p}$ increases the parametric robustness of the design around $\mathbf{p}$. One may then consider the following trade-off between nominal performance and robustness:

$$(5) \qquad \begin{array}{ll} \text{minimize} & \mathcal{R}(\mathbf{p}_0, \mathbf{x}) \\ \text{subject to} & \mathcal{P}(\mathbf{p}_0, \mathbf{x}) \leq p^{\text{nom}}(1+\alpha) \\ & K(\mathbf{x}) \text{ internally stabilizing} \end{array}$$

Denoting the solution of (5) as $\mathbf{x}^{\text{rob}}$, we can roughly say that the robust controller $K(\mathbf{x}^{\text{rob}})$ accepts a loss of $\alpha \cdot 100\%$ over nominal performance p^{nom} and uses this new freedom to buy some additional robustness.

Several robustness measures are known in the literature. A classical idea is to use the various sensitivity functions, see e.g. [5]. Here we propose a new idea, which uses the variation of $\mathcal{P}$ directly to robustify program (4):

$$\mathcal{R}(\mathbf{p}, \mathbf{x}) = \|\nabla_{\mathbf{p}} \mathcal{P}(\mathbf{p}, \mathbf{x})\|^2,$$

where $\|\cdot\|$ denotes the euclidean norm in parameter space.

3.1 Computing $\mathcal{R}(G, K)$

Assuming without loss that $G = G(\mathbf{p}_0)$ is augmented and K is static, we put

$$\mathcal{A}(G,K) = A + BKC, \quad \mathcal{B}(G,K) = B_2 + BKD_{21},$$
$$\mathcal{C}(G,K) = C_2 + D_{12}KC, \quad \mathcal{D}(G,K) = D_{12}KD_{21} = 0.$$

Then the squared H_2 norm can be expressed as

$$(6) \qquad \mathcal{P}(G,K) = \operatorname{Tr}\left(\mathcal{B}(K)^\top X \mathcal{B}(K)\right) = \operatorname{Tr}\left(\mathcal{C}(K) Y \mathcal{C}(K)^\top\right),$$

where $X = X(G,K)$ is solution of

$$(7) \qquad \mathcal{A}(G,K)^\top X + X\mathcal{A}(G,K) + \mathcal{C}(G,K)^\top \mathcal{C}(G,K) = 0,$$

and $Y = Y(G,K)$ is solution of

$$(8) \qquad \mathcal{A}(G,K) Y + Y \mathcal{A}(G,K)^\top + \mathcal{B}(G,K)\mathcal{B}(G,K)^\top = 0.$$

This allows to compute partial derivatives of $\mathcal{P}$ with respect to G and K.

Lemma 1. *The objective $\mathcal{P}$ in (6) is smooth in the open domain of all closed-loop stabilizing pairs (G,K). For any (G,K) in this set we have*

1. $\nabla_K \mathcal{P}(G,K) = 2\left[B^\top X + D_{12}^\top \mathcal{C}(K)\right] Y C^\top + 2B^\top X \mathcal{B}(K) D_{21}^\top,$
2. $\nabla_A \mathcal{P}(G,K) = 2XY,$
3. $\nabla_B \mathcal{P}(G,K) = 2XYC^\top K^\top + 2X\mathcal{B}(K)D_{21}^\top K^\top.$
4. $\nabla_C \mathcal{P}(G,K) = 2K^\top B^\top XY + 2K^\top D_{12}^\top \mathcal{C}(K) Y,$
5. $\nabla_{C_2} \mathcal{P}(G,K) = 2\mathcal{C}(K)Y,$
6. $\nabla_{B_2} \mathcal{P}(G,K) = 2X\mathcal{B}(K),$
7. $\nabla_{D_{21}} \mathcal{P}(G,K) = 2K^\top B^\top X \mathcal{B}(K),$
8. $\nabla_{D_{12}} \mathcal{P}(G,K) = 2Y^\top C^\top K^\top,$

where X solves (7) and Y solves (8).

The proof will be sketched in the appendix. Recall that we are dealing with structured controllers. Smooth dependence on $\mathbf{x}$ allows an expansion of the form $K(\mathbf{x}) = K(\mathbf{x}_0) + \sum_{i=1}^n K_i(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_0\|^2)$, where $K_i(\mathbf{x}_0) = \frac{\partial K(\mathbf{x}_0)}{\partial \mathbf{x}_i}$. Using the chain rule, we get

Corollary 1. *Under the assumptions of Lemma 1 we have $\nabla_x \mathcal{P}(\mathbf{x}, \mathbf{p}) = (g_1(\mathbf{p}, \mathbf{x}), \dots, g_n(\mathbf{p}, \mathbf{x}))$, where $g_i(\mathbf{p}, \mathbf{x}) =$*

$$\operatorname{Tr}\left[\left(2\left[B^\top X + D_{12}^\top \mathcal{C}(K)\right] YC^\top + 2B^\top X \mathcal{B}(K) D_{21}^\top\right)^\top K_i(\mathbf{x})\right].$$

□

Let us now specialize to the case where only the system matrix A in G features uncertain parameters $\mathbf{p}$. The general case, where uncertain parameters appear in other parts of G, can be handled analogously. Assuming a smooth dependence on $\mathbf{p}$, we get an expansion of the form $A(\mathbf{p}) = A(\mathbf{p}_0) + \sum_{i=1}^s A_i(\mathbf{p}_0)(\mathbf{p} - \mathbf{p}_0) + \mathcal{O}(\|\mathbf{p} - \mathbf{p}_0\|^2)$, where $A_i(\mathbf{p}_0) = \frac{\partial A(\mathbf{p}_0)}{\partial \mathbf{p}_i}$. We have the following

Corollary 2. *Under the assumptions of Lemma 1 we have: $\nabla_p \mathcal{P}(\mathbf{p}, \mathbf{x}) = (h_1(\mathbf{p}, \mathbf{x}), \dots, h_s(\mathbf{p}, \mathbf{x}))$, where $h_i(\mathbf{p}, \mathbf{x}) = 2\operatorname{Tr}(A_i(\mathbf{p})^\top XY)$.* □

Smallness of the variation $\nabla_p \mathcal{P}(\mathbf{p}_0, \mathbf{x})$ at the solution $K(\mathbf{x})$ can be assessed by controlling its size in some norm. If a norm $\|\mathbf{p}\|$ in parameter space is given, reflecting for instance an appropriate weighting between the uncertain parameters, then we are led to control $\nabla_p \mathcal{P}$ in the dual norm $\|\cdot\|_*$. During the following we shall consider the Euclidean norm $\|\mathbf{p}\|$, so that $\|\cdot\|_*$ is also the Euclidean norm. (The reader will easily see how to extend our approach to other choices of $\|\cdot\|$.) With these arrangements our robustness objective should be chosen as

$$(9) \qquad \mathcal{R}(\mathbf{p}_0, \mathbf{x}) = \|\nabla_p \mathcal{P}(A(\mathbf{p}_0), K(\mathbf{x}))\|_2^2 = \sum_{i=1}^s \operatorname{Tr}\left(2A_i(\mathbf{p}_0)^\top XY\right)^2 = \sum_{i=1}^s h_i(\mathbf{p}_0, \mathbf{x})^2.$$

3.2 Computing $\nabla_x \mathcal{R}(\mathbf{p}, \mathbf{x})$

This seems to indicate that almost no extra work is needed for the new robustness function, but the question is how to compute derivatives of $\mathcal{R}$ with respect to $\mathbf{x}$. We have

$$\nabla_x \mathcal{R}(\mathbf{p}, \mathbf{x}) = \sum_{i=1}^{s} h_i(\mathbf{p}, \mathbf{x}) \nabla_x h_i(\mathbf{p}, \mathbf{x}),$$

where the h_i are given in Corollary 2 and are readily computed from X, Y. We can therefore concentrate on how gradients $\nabla_x h_i$ are computed. We recognize this as a matrix realization of the mixed second derivative $D^2_{x,p}\mathcal{P}$. Unfortunately, unlike first-order derivatives, it is not clear how to compute matrix representations at the second order level. In [13] a representation of the Hessian $\nabla^2_{KK}\mathcal{P}$ is obtained, but closer inspection shows that Kronecker products are used and matrix inversions are required. Here we favour an approach where parts of the mixed second derivative are pre-calculated, while the rest is computed on the fly. There are two possibilities to represent $D^2_{x,p}\mathcal{P}$, namely, $D_p\nabla_x\mathcal{P}$ or $D_x\nabla_p\mathcal{P}$. In the case where $\dim(\mathbf{p}) < \dim(\mathbf{x})$ we compute $D_p\nabla_x\mathcal{P}$. We have

$$\begin{aligned}
\langle \nabla_x h_i(\mathbf{x}, \mathbf{p}_0), \Delta x\rangle &= D_x h_i(\mathbf{x}, \mathbf{p}_0)\Delta x \\
&= D_x D_p \mathcal{P}(\mathbf{x}, \mathbf{p}_0)\Delta \mathbf{p}_i \Delta \mathbf{x} \\
&= \langle D_{p_i}\nabla_x \mathcal{P}(\mathbf{x}, \mathbf{p}_0), \Delta x\rangle \\
&= \langle D_A \nabla_x \mathcal{P}(A(\mathbf{p}_0), K(\mathbf{x})) A_i(\mathbf{p}_0), \Delta \mathbf{x}\rangle \\
&= \sum_{k=1}^{n} \langle D_A \nabla_K \mathcal{P}(A(\mathbf{p}_0), K(\mathbf{x})) A_i(\mathbf{p}_0), K_k(\mathbf{x})\rangle \Delta \mathbf{x}_k.
\end{aligned}$$

Substituting the expression in item 1 of Lemma 1 for $\nabla_K \mathcal{P}$, we get

$$\begin{aligned}
D_A \nabla_K \mathcal{P}(A(\mathbf{p}_0), K(\mathbf{x})) A_i(\mathbf{p}_0) &= 2B^\top \Phi_i Y C^\top \\
&\quad + 2[B^\top X + D_{12}^\top \mathcal{C}(K(\mathbf{x}))]\Psi_i C^\top \\
&\quad + 2B^\top \Phi_i \mathcal{B}(K(\mathbf{x}) D_{21}^\top,
\end{aligned}$$

where

$$\Phi_i = D_A X A_i(\mathbf{p}_0), \quad \Psi_i = D_A Y A_i(\mathbf{p}_0), \quad i = 1, \ldots, s.$$

Then, putting

$$(10) \quad \Lambda_i = 2B^\top \Phi_i Y C^\top + 2[B^\top X + D_{12}^\top \mathcal{C}(K(\mathbf{x}))]\Psi_i C^\top + 2B^\top \Phi_i \mathcal{B}(K(\mathbf{x}) D_{21}^\top,$$

$i = 1, \ldots, s$, and $\Lambda = \sum_{i=1}^{s} h_i(\mathbf{x}, \mathbf{p}_0)\Lambda_i$, we obtain the gradient $\nabla_x \mathcal{R}$ as

$$\nabla_x \mathcal{R}(\mathbf{x}) = \Big(\mathrm{Tr}(\Lambda^\top K_1(\mathbf{x})), \ldots, \mathrm{Tr}(\Lambda^\top K_n(\mathbf{x}))\Big).$$

The final link is now to compute Φ_i and Ψ_i, which requires another set of Lyapunov equations. We have the following

Proposition 1. *Computing $\mathcal{R}(\mathbf{p}_0, \mathbf{x})$ and its gradient $\nabla_x \mathcal{R}(\mathbf{p}_0, \mathbf{x})$ with respect to $\mathbf{x}$ is possible by solving $2(s+1)$ Lyapunov equations. Those are (7) for X, (8) for Y,*

$$(11) \quad [A + BK(\mathbf{x})C]^\top \Phi_i + \Phi_i [A + BK(\mathbf{x})C] = -A_i(\mathbf{p}_0)^\top X - X A_i(\mathbf{p}_0)$$

for the Φ_i, $i = 1, \ldots, s$, and

$$(12) \quad [A + BK(\mathbf{x})C]\Psi_i + \Psi_i [A + BK(\mathbf{x})C]^\top = -Y A_i(\mathbf{p}_0)^\top - A_i(\mathbf{p}_0) Y$$

for the Ψ_i, $i = 1, \ldots, s$. □

We have the following

Algorithm to compute $\mathcal{R}$ and its gradient $\nabla_x \mathcal{R}$

Parameters: Precomputed data $A_i = \frac{\partial A(\mathbf{p}_0)}{\partial \mathbf{p}_i}$ and possibly $K_\nu = \frac{\partial K(\mathbf{x})}{\partial \mathbf{x}_\nu}$.

1: Given $\mathbf{x}$ compute $K = K(\mathbf{x})$, solution X of (7), and solution Y of (8).
2: For $i = 1, \ldots, s$ compute $A_i^\top XY$ and $\mathcal{R}$ using (9).
3: For $i = 1, \ldots, s$ compute Φ_i solution of (11), and Ψ_i solution of (12).
4: Let $h(\mathbf{p}_0, \mathbf{x}) = \left(\mathrm{Tr}\left(2A_1^\top XY\right), \ldots, \mathrm{Tr}\left(2A_s^\top XY\right)\right)$ according to Corollary 2.
5: For $i = 1, \ldots, s$ compute Λ_i according to (10). Then compute $\Lambda = \sum_{i=1}^{s} h_i \Lambda_i$.
6: If $K(\mathbf{x})$ is not affine then compute $K_\nu(\mathbf{x})$. Otherwise take the precomputed K_ν.
7: Obtain $\nabla_x \mathcal{R} = \left(\mathrm{Tr}(\Lambda^\top K_1(\mathbf{x})), \ldots, \mathrm{Tr}(\Lambda^\top K_n(\mathbf{x}))\right)$.

4 Numerical Experiment

4.1 Benchmark Example

We consider the mass-spring system in Figure 1, which can be considered as a prototype of a flexible system.

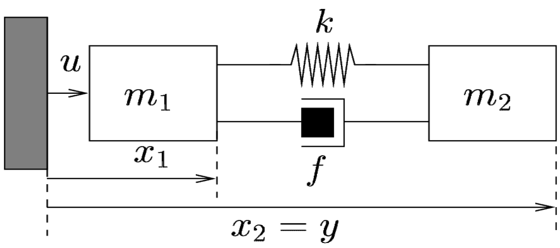

FIGURE 1: Mass-spring system. Nominal data are $m_1 = m_2 = 0.5$kg, $k = 1$N/m, $f = 0.0025$Ns/m, $V = W = 1$. Measured output is $y = x_2$, control force u acts on m_1.

We perform an LQG study where we expect the LQG controller to be robustly stable with respect to 30% variation in m_2 and k. The LQG set-up has $W = BB^T$, $V = I$, $Q = C^T C$, $R = I$ and is as usual transformed to a standard H_2 plant (3). The data are

$$(13) \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{f}{m_1} & \frac{k}{m_1} & \frac{f}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k}{m_2} & \frac{f}{m_2} & \frac{-k}{m_2} & \frac{-f}{m_2} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ \frac{1}{m_1} \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}, D = 0.$$

Since an observer-based controller (2) is of order $n_K = 4$, we have to augment the system from $A \in \mathbb{R}^{4\times 4}$ to $A^{\mathrm{aug}} \in \mathbb{R}^{8\times 8}$, as in section 2.3. The non-linear expression $A(\mathbf{p}) = A(\mathbf{p}_0 + \Delta\mathbf{p})$ is

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k+\Delta k}{m_1} & -\frac{f}{m_1} & \frac{k+\Delta k}{m_1} & \frac{f}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k+\Delta k}{m_2+\Delta m_2} & \frac{f}{m_2+\Delta m_2} & \frac{-k-\Delta k}{m_2+\Delta m_2} & \frac{-f}{m_2+\Delta m_2} \end{bmatrix}$$
$$= A(\mathbf{p}_0) + D_p A(\mathbf{p}_0)\Delta\mathbf{p} + \mathcal{O}(\|\Delta\mathbf{p}\|^2),$$

which gives us $D_p A(\mathbf{p}_0)\Delta\mathbf{p} =$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{\Delta k}{m_1} & 0 & \frac{\Delta k}{m_1} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{m_2\Delta k - k\Delta m_2}{m_2^2} & \frac{-f\Delta m_2}{m_2^2} & \frac{-m_2\Delta k + k\Delta m_2}{m_2^2} & \frac{f\Delta m_2}{m_2^2} \end{bmatrix}.$$

$$A_1(\mathbf{p}) = \frac{\partial A}{\partial k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\frac{1}{m_1} & 0 & \frac{1}{m_1} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m_2} & 0 & -\frac{1}{m_2} & 0 \end{bmatrix},$$

$$A_2(\mathbf{p}) = \frac{\partial A}{\partial m_2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{k}{m_2^2} & -\frac{f}{m_2^2} & \frac{k}{m_2^2} & \frac{f}{m_2^2} \end{bmatrix}.$$

Putting $Z = 2YX$, we obtain $h_1(\mathbf{p},\mathbf{x}) = \mathrm{Tr}(ZA_1) = Z_{32}/m_1 + Z_{34}/m_2 - Z_{12}/m_1 + Z_{14}/m_2$ and $h_2(\mathbf{p},\mathbf{x}) = -kZ_{14}/m_2^2 - fZ_{24}/m_2^2 + kZ_{34}/m_2^2 + fZ_{44}/m_2^2$. □

4.2 Results

As can be seen in Figure 2 top, the nominal LQG controller $K_{\mathrm{nom}} = K(K_c^{\mathrm{nom}}, K_f^{\mathrm{nom}})$ misses this goal. Program (5) with (9) is used to enhance parametric robustness of the nominal controller. The result is $K_{\mathrm{rob}} = K(K_c^{\mathrm{rob}}, K_f^{\mathrm{rob}})$ and its parametric robustness is shown in Figure 2 middle. Notice that in program (5) the observer structure has to be imposed as a constraint. As a curiosum, no algebraic Riccati equations are obtained for $K_c^{\mathrm{rob}}, K_f^{\mathrm{rob}}$, but the observer structure is nevertheless maintained. Robustness leads to a degradation of nominal performance from $\mathcal{P}(G, K_{\mathrm{nom}}) = 3.99$ to $\mathcal{P}(G, K_{\mathrm{rob}}) = 27.98$.

A classical method to enhance robustness of LQG is the LTR procedure, which we applied here for the purpose of comparison to the input sensitivity function. This generates a family $K(\rho)$ of LQG controllers based on modified plants $G(\rho)$, where $\rho = 0$ corresponds to the nominal case G. As ρ increases, the stability region of $K(\rho)$ increases, while $\mathcal{P}(G, K(\rho))$ degrades. In this study LTR was unable to achieve parametric robustness over the square of 30% parameter variations. Figure 2 (bottom) shows the stability region of $K_{\mathrm{ltr}} := K(\rho)$, adjusted so that $\mathcal{P}(G, K_{\mathrm{ltr}}) = 27.98$.

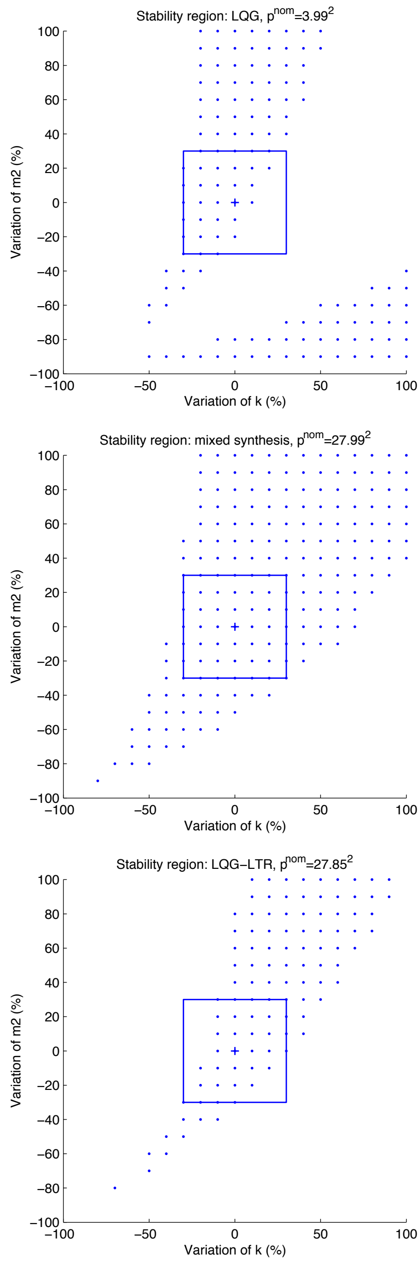

FIGURE 2: Stability region of LQG controller (top), robust LQG controller based on (5) (middle), and LQG/LTR controller (bottom). The value $\alpha = 45$ is used to compute the robust LQG controller. Robust and LTR controller have the same nominal performance.

Notice another curiosity: the LTR controller is also observer-based with $K_f^{\rm ltr}, K_c^{\rm ltr}$ now satisfying algebraic Riccati equations, but the wrong ones so to say, because $G(\rho)$ replaces G.

In Fig. 3 the relative performance $\frac{\mathcal{P}(G(k,m_2),K)-\mathcal{P}(G(k^0,m_2^0,K)}{\mathcal{P}(G(k^0,m_2^0),K)} \times 100\%$ is plotted over the square $\Omega = (k^0 \pm 30\% k^0, m_2^0 \pm 30\% m_2^0)$ and for $K \in \{K_{\rm nom}, K_{\rm rob}, K_{\rm ltr}\}$. For $K_{\rm lqg} = K_{\rm nom}$ this value is not finite everywhere and reaches 600% in the region where the system is still stabilized. In contrast, the robustified LQG controller $K_{\rm rob}$ holds a fairly uniform performance level over the entire square (less than 1% variation), but performs worse at the nominal parameter value $\mathbf{p}_0$. To compare (5) with the LQG/LTR procedure, the stability domain is compared for two controllers achieving the same performance $\mathcal{P} = 27.98$ at $\mathbf{p}_0$.

5 Conclusion

Lack of parametric robustness of LQG controllers and more general structured H_2 controllers was addressed by a constrained program (5), which accepts a quantified loss of nominal performance in order to gain additional robustness. We proposed to use a suitable norm of the variation of the performance criterion as a robustness index. In the context of LQG the new procedure was compared to the LQG/LTR procedure based on the input sensitivity function, which is a classical procedure to enhance system robustness.

6 Appendix

The first item follows readily from [13, Theorem 3.2]. We elaborate on items 2. - 8. For a function $\mathcal{P} : H_1 \times H_2 \to \mathbb{R}$, where H_1, H_2 are Hilbert spaces, we let $D_x\mathcal{P}(x,y)$ denote the partial derivative with respect to $x \in H_1$, which is a continuous linear functional on H_2. The gradient $\nabla_x \mathcal{P}(x,y) \in H_2$ is related to $D_x\mathcal{P}(x,y)$ by $D_x\mathcal{P}(x,y)\Delta y = \langle \nabla_x \mathcal{P}(x,y), \Delta y\rangle$ for every $\Delta y \in H_2$. Notice that

$$D_G\mathcal{P}\Delta G = \text{Tr}\left(\{D_G X \Delta G\}\mathcal{B}\mathcal{B}^\top\right) + 2\text{Tr}\left(X\{D_G\mathcal{B}\Delta G\}\mathcal{B}^\top\right),$$

omitting arguments, where $\Phi := D_G X\Delta G$ solves the Lyapunov equation

$$\begin{aligned}(14)\ \mathcal{A}^\top\Phi + \Phi\mathcal{A} = &-\{\Delta_G\mathcal{A}\Delta G\}^\top X - X\{\Delta_G\mathcal{A}\Delta G\} \\ &-\{D_G\mathcal{C}\Delta G\}^\top\mathcal{C} - \mathcal{C}^\top\{D_G\mathcal{C}\Delta G\}.\end{aligned}$$

We multiply (14) with Y from the right, and match it with (8) multiplied with Φ from the left. Taking traces, the two left hand sides are identical, hence the same is true for the two right hand sides. This gives the identity $\text{Tr}\left(\Phi\mathcal{B}\mathcal{B}^\top\right) = 2\text{Tr}\left(\{D_G\mathcal{A}\Delta G\}^\top XY\right) + 2\text{Tr}\left(\{D_G\mathcal{C}\Delta G\}^\top\mathcal{C}Y\right)$. Substituting this back in the formula for $D_G\mathcal{P}\Delta G$ gives $D_G\mathcal{P}\Delta G = 2\text{Tr}\left(\{D_G\mathcal{A}\Delta G\}^\top XY\right) + 2\text{Tr}\left(\{D_G\mathcal{C}\Delta G\}^\top\mathcal{C}Y\right) + 2\text{Tr}\left(X\{D_G\mathcal{B}\Delta G\}\mathcal{B}^\top\right)$.
Now observe that

$$\begin{aligned}D_G\mathcal{A}(G,K)\Delta G &= \Delta A + \Delta BKC + BK\Delta C,\\ D_G\mathcal{C}(G,K)\Delta G &= \Delta C_2 + \Delta D_{12}KC + D_{12}K\Delta C,\\ D_G\mathcal{B}(G,K)\Delta G &= \Delta B_2 + \Delta BKD_{21} + BK\Delta D_{21}.\end{aligned}$$

Hence

$$\begin{aligned}\langle\nabla_G\mathcal{P}(G,K),\Delta G\rangle = \ &\text{Tr}\left((\Delta A + \Delta BKC + BK\Delta C)^\top 2XY\right)\\ &+\text{Tr}\left((\Delta C_2 + \Delta D_{12}KC + D_{12}K\Delta C)^\top 2(C_2 + D_{12}KC)Y\right)\\ &+\text{Tr}\left(2X(\Delta B_2 + \Delta BKD_{21} + BK\Delta D_{21})\mathcal{B}^\top\right).\end{aligned}$$

From that we can readily read off the answers 2. - 8., bearing in mind that

$$\langle\nabla_G\mathcal{P},\Delta G\rangle = \langle\nabla_A\mathcal{P},\Delta A\rangle + \cdots + \langle\nabla_{D_{12}}\mathcal{P},\Delta D_{12}\rangle.$$

□

Acknowledgement

This work was supported by research grants *Technicom* from the Fondation d'Entreprise EADS and *Survol* from Fondation de Recherche pour l'Aéronautique et l'Espace.

References

[1] P. Apkarian, L. Ravanbod-Hosseini, D. Noll. *Time domain constrained H_∞ synthesis.* Int. J. of Rob. and Nonlin. Contr., in press.

[2] R. P. Braatz, P. M. Young, J. C. Doyle, M. Morari. *Computational complexity of μ calculations.* IEEE Transactions on Automatic Control, vol. 39, no. 5, pp. 1000 – 1002.

[3] J. W. Demmel. *The component-wise distance to the nearest singular matrix.* SIAM Journal on Matrix Analysis and Applications, vol. 13, 1992, pp. 10 – 19.

[4] J.C. Doyle. *Structured uncertainty in control system design.* Proc. 24th Conference on Decision and Control, Ft. Lauderdale, Fl, 1985, pp. 260 – 265.

[5] J. Doyle, K. Glower, P.P. Kargonekar, B. Francis. *State-space solutions to the standard H_2 and H_∞ control problems.* IEEE Trans. Autom. Control, vol. 34, no. 8, 1989, pp. 831 – 847.

[6] J. Doyle, A. Packard, K. Zhou. *Review of LTS's, LMI's, and μ.* Proceedings of the IEEE Conference on Decision and Control, vol. 2, Brighton 1991, pp. 1227 – 1232.

[7] J. Doyle, G. Stein. *Robustness with observers*, IEEE Transactions on Autom. Control, vol. AC-24, 1979, pp. 607 – 611.

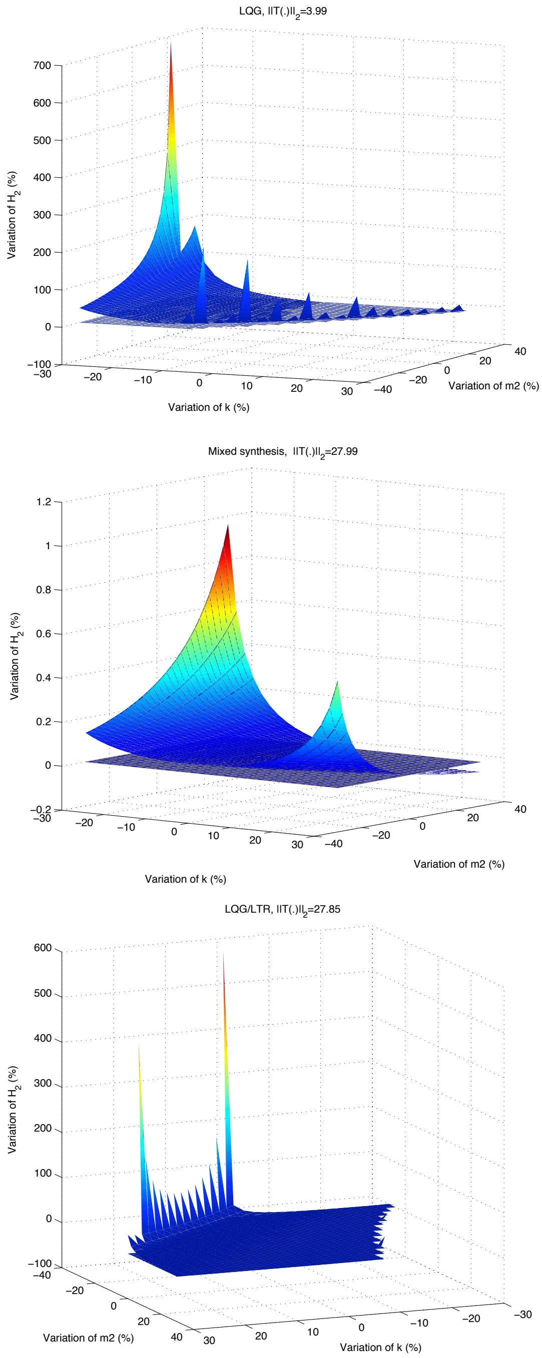

Figure 3: Relative performance of K_{lqg}, K_{rob}, K_{ltr} is plotted over the robustness square. Upper graph shows LQG controller, middle image shows robust controller based on (5), lower image shows LQG/LTR controller.

[8] J. Doyle, G. Stein. *Multivariable feedback design: concepts for a classical/modern synthesis.* IEEE Transactions on Automatic Control, vol. AC-26, no. 1, 1981, pp. 4 – 16.

[9] J. Doyle. *Guaranteed margins for LQG regulators.* IEEE Trans. on Auto. Control, vol. 23, no.4, 1978, pp. 756-757.

[10] C.-M. Fransson, T. Wik, B. Lennartson, M. Saunders, P.-O. Gutman. *Nonconservative robust control: optimized and constrained sensitivity functions.* IEEE Transactions on Control System Technology, vol. 17, no. 2, 2009, pp. 298 – 308.

[11] H. Kwakernaak. *Optimal low-sensitivity linear feedback systems.* Automatica, vol. 5, 1969, pp. 279 – 285.

[12] S. Poljak, J. Rohn. *Checking robust nonsingularity is NP-hard.* Math. Contr. Signals Syst., vol. 6, 1993, pp. 1 – 9.

[13] T. Rautert, E. Sachs. *Computational design of optimal output feedback controllers.* SIAM J. Optim., vol. 7, no. 3, 1997, pp. 837 – 852.

[14] A. Saberi, Ben M. Chen, P. Sannuti. *Loop transfer recovery: analysis and design.* Communication and Control Engineering Series, Springer Verlag 1993.

[15] K. Zhou. Essentials of robust control. Prentice Hall, 1998.