GAIN-SCHEDULED PID FOR IMBALANCE COMPENSATION OF A MAGNETIC BEARING

Keywords: Scheduled controller for magnetic bearing, robust control, distance to instability

Abstract: Control of a magnetic bearing device is addressed by parameter varying control. We compare linearly interpolated controllers and a switching strategy with and without hysteresis. Piecewise LPV controllers are found to be an interesting alternative.

1 INTRODUCTION

Magnetic bearings (MB) increasingly become the choice for high-speed, high-performance rotating machinery because of their frictionless characteristics. They utilize a magnetic field generated by radially or axially placed electromagnets to generate the forces necessary to suspend and support a shaft without any contact with its environment. Thus, magnetic bearings are particularly useful in very high or very low temperature conditions where a lubrication-free environment is necessary. The advantages of magnetic bearings are primarily their very low power consumption and their very long maintenance-free life. Some applications where magnetic bearings offer distinct advantages are high speed turbo machinery, precision milling spindles, and combined attitude control and energy storage for spacecraft and satellites. A disadvantage of magnetic bearings is that they require continuous power input and active control to hold the load stable.

Active magnetic bearings (AMB) can support rotors without friction but require a sophisticated control system because specific performance requirements such as automatic balancing of the shaft, rejection of unwanted disturbances and vibration isolation are required. Many of the controllers proposed assume a linear time-invariant model, an assumption which is no longer accurate when the rotational speed varies.

Control techniques from linear robust control (Mohamed and Busch-Vishniac, 1995), H_∞ loop shaping and μ synthesis (Lanzon and Tsiotras, 2005) as well as adaptive control methods have been used to attack this problem. Robust control is often overly conservative as it fails to account for the actual time variation of the rotor speed, which in addition is measurable.

Another approach is gain-scheduled H_∞ controllers for linear parameter varying (LPV) systems based on LMI techniques (Tsiotras and Mason, 1996; Packard, 1994; Apkarian and Gahinet, 1996; Apkarian et al., 1995). Here the idea is to solve a series of standard H_∞ problems at a pre-specified number of operating speeds. Using a single Lyapunov function to show stability and finite L_2-gain at these selected points, one guarantees that these properties will also hold for all operating speeds which are linear combinations of the selected speeds (interpolation). Unfortunately, this strategy is only valid if the controller is of the same order that system, and due to the choice of a single Lyapunov functions also tends to be conservative. In AMB systems there is strong interest to use small order controllers or other simple structures like PID.

High rotor speeds are gaining importance, and the fast sampling rate necessary for these MB systems makes the application of digital control a difficult task. Fast sampling rates call for simplification of feedback matrices in control design. In order to comply with the demand for simplicity, our present study uses PID controllers.

PID are still the controllers of choice due to consolidated hardware and software tools for design and hardware embedding, and the implication that more complex controllers may have. The drawback of PIDs in scheduling may be a significant lost of performance, and sometimes even worse, loss of internal stability. Here we present a new method to design scheduled PIDs for a parameter varying MB system, which allows to avoid these fallacies.

The paper is organized as follows. Section 2 represents the open-loop MB system. In section 3 the H_∞ performance channel is discussed, and section 4 gives the state-space form of the decentralized PID. Section 5 explains the role of the reference curve $K^*(p)$ and explains the rationale of the robustification method. Re-centralizing plant and controller models of MB are explained in sections 5.1 and 5.2 and are needed to apply the LPV procedure in sections 6 and 7. This is carried out in section 7 by solving a mixed H_∞/H_∞ program (11) based on the semi-structured stability radius (see (D. Hinrichsen, 1986a; D. Hinrichsen, 1986b; M. Karow, 2010; Lawrence et al., 2000)). Experimental results are presented in section 8.

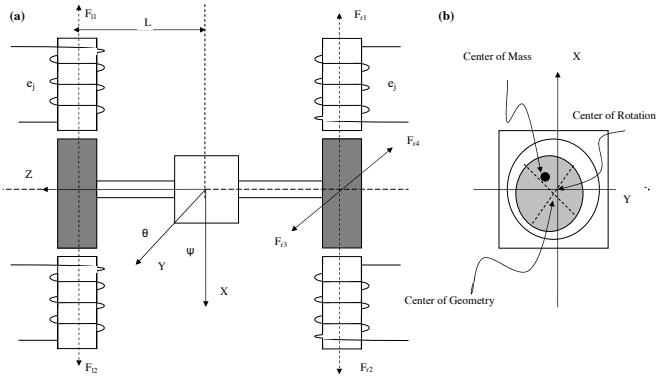

Figure 1: Left image, (a), shows the magnetic bearing configuration with magnetic forces $F_{r1},\ldots,F_{r4}$, $F_{l1},\ldots,F_{l4}$ according to (R.D. Smith, 1995). Right image, (b), shows the rotor unbalance. The purpose of unbalance compensation is to drive the displacements $x_1 = L\theta$ and $x_2 = L\psi$ to 0.

2 OPEN LOOP

The open-loop magnetic bearing system has the following parameter dependent form

$$\begin{array}{rcl} \dot{x}_{\rm mb} & = & A_{\rm mb}(p)x_{\rm mb} + B_{\rm mb}u \\ y & = & C_{\rm mb}x_{\rm mb} + w \end{array} \qquad (1)$$

where $x_{\rm mb} \in \mathbb{R}^6$, while $w,u,y \in \mathbb{R}^2$ and the state-space matrices are given by:

$$A_{\rm mb}(p) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ \frac{-4c_2}{m} & 0 & 0 & \frac{-pJ_\alpha}{J_r} & \frac{2c_1}{m} & 0 \\ 0 & \frac{-4c_2}{m} & \frac{pJ_\alpha}{J_r} & 0 & 0 & \frac{2c_1}{m} \\ \frac{2d_2}{N} & 0 & 0 & 0 & \frac{-2d_1}{N} & 0 \\ 0 & \frac{2d_2}{N} & 0 & 0 & 0 & \frac{-2d_1}{N} \end{bmatrix},$$

$$B_{\rm mb} = \frac{1}{N}\begin{bmatrix} 0_{4\times 2} \\ I_2 \end{bmatrix}, \quad C_{\rm mb} = [I_2 \;\; 0_{2\times 4}]$$

with parameters gathered in Table 1. The exogenous input $w = [w_1, w_2]^T$ is a sinusoidal sensor disturbance of the form $w_1 = \tilde{d}e^{-\phi t}\cos(pt+\eta)$ and $w_2 = \tilde{d}e^{-\phi t}\sin(pt+\eta)$, which models the unbalance of the bearing. Here $\tilde{d}$ is the magnitude of the unbalance and η corresponds to un unknown initial phase angle. The varying parameter p represents the rotor velocity, which is measured on-line and varies in the range $p \in \Pi := [315, 1100]$ rad/s. One can represent the sensor disturbance by the following state space representation:

$$\begin{array}{rcl} \dot{x}_{\rm dist} & = & A_{\rm dist}(p)x_{\rm dist} + B_{\rm dist}\tilde{d} \\ w & = & C_{\rm dist}x_{\rm dist} \end{array} \qquad (2)$$

where

$$A_{\rm dist}(p) = \begin{bmatrix} -2\phi & -p \\ p & 0 \end{bmatrix}, B_{\rm dist} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_{\rm dist} = I_2,$$

and $\phi = 0.05$. The system is therefore described by the joint system state $x_{\rm sys} = [x_{\rm mb}, x_{\rm dist}]^T$ with dynamics

$$\begin{array}{rcl} \dot{x}_{\rm sys} & = & A_{\rm sys}(p)x_{\rm sys} + B_{1\,\rm sys}\tilde{d} + B_{2\,\rm sys}u \\ y & = & C_{\rm sys}x_{\rm sys} \end{array} \qquad (3)$$

where $A_{\rm sys}(p) = \mathrm{diag}(A_{\rm mb}(p), A_{\rm dist}(p)) \in \mathbb{R}^{8\times 8}$, $B_{1\,\rm sys} = [0_{6\times 1},\; B_{\rm dist}]^T$, $B_{2\,\rm sys} = [B_{\rm mb},\; 0_{2\times 2}]$, $C_{\rm sys} = [C_{\rm mb},\; C_{\rm dist}]$. For more explication on system modeling see (Lanzon and Tsiotras, 2005), (Tsiotras and Mason, 1996) and (R.D. Smith, 1995). In the latter reference a more complete model is presented, in which p is one of the states. This means that we expect p to vary continuously in time.

3 PERFORMANCE INDEX

In our next step we have to define controlled outputs $z \in \mathbb{R}^4$ to assess the system performance. We use the control configuration shown in Figure 2. Following (Tsiotras and Mason, 1996), the controlled output z regroups u and y with appropriate frequency weighing filters: $W_u = \mathrm{diag}(W_{u1}, W_{u2})$ with $W_{u1} =$

c_1	$1.9715\,e^5\ Wb$	N	400
c_2	$325.047\ Wb^2/m$	J_α	$0.0136\,Kg.m^2$
d_1	$2.1001\,e^4\ \Omega.Wb/H.m$	J_r	$0.333\,Kg.m^2$
d_2	$7.9804\,e^3\ \Omega.Wb/H.m$	m	$19.7041\,Kg$

Table 1: System constants

$W_{u2} = 0.01\frac{(s+1500)^2}{(s+10000)^2}$ and $W_y = \mathrm{diag}(0.5, 0.5)$ which up to a factor 10^{-4} are as in (Tsiotras and Mason, 1996). Altogether this adds 4 states to the 8 states of the open loop system. We add a reference signal

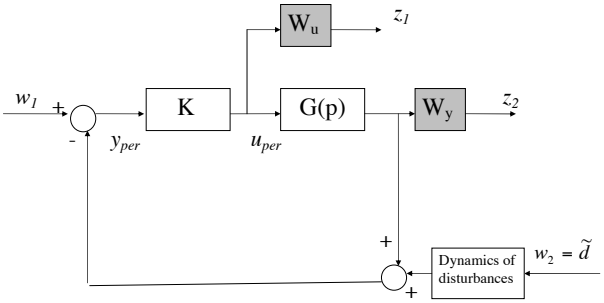

Figure 2: Block diagram of H_∞ system.

$r \in \mathbb{R}^2$ for y to the exogenous inputs, which leads to $w = (r_1, r_2, \widetilde{d}) \in \mathbb{R}^3$. Altogether we obtain a parameter varying LFT of the form

$$P(p): \left[\begin{array}{c} \dot{x} \\ \hline z_{\mathrm{per}} \\ y \end{array}\right] = \left[\begin{array}{c|cc} A(p) & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline w_{\mathrm{per}} \\ u \end{array}\right]$$

with dimensions $x \in \mathbb{R}^{12}$, $z_{\mathrm{per}} \in \mathbb{R}^4$, $y \in \mathbb{R}^2$, $w_{\mathrm{per}} \in \mathbb{R}^3$ and $u \in \mathbb{R}^2$. We shall refer to $w_{\mathrm{per}} \to z_{\mathrm{per}}$ as the performance channel.

4 CONTROLLER PARAMETRIZATION

In this study we design a decentralized PID controller which depends on the scheduling parameter p. A state-space representation is of the form

$$K(p) = \left[\begin{array}{cccc|cc} 0 & 0 & 0 & 0 & r_i(p) & 0 \\ 0 & \tau(p) & 0 & 0 & r_d(p) & 0 \\ 0 & 0 & 0 & 0 & 0 & r_i'(p) \\ 0 & 0 & 0 & \tau'(p) & 0 & r_d'(p) \\ \hline 1 & 1 & 0 & 0 & d(p) & 0 \\ 0 & 0 & 1 & 1 & 0 & d'(p) \end{array}\right] \quad (4)$$

with 8 scheduling functions $r_i(p), \ldots, r_d'(p)$ to be determined. If we assume an affine parametrization, then we have 16 free parameters to determine.

5 RATIONALE

For every parameter value p we consider the H_∞-synthesis problem

$$\begin{array}{ll} \text{minimize} & \|T_{w_{\mathrm{per}} \to z_{\mathrm{per}}}(P(p), K)\|_\infty \\ \text{subject to} & K \text{ has decentralized PID structure (4)} \\ & K \text{ stabilizes } P(p) \text{ internally} \end{array} \quad (5)$$

Let $K^*(p)$ be the solution of (5), which we compute by the Matlab function `hinfstruct`. This furnishes 8 optimal parameter values $r_i^*(p), \ldots, d'^*(p)$. The curve $p \mapsto \|T_{w_{\mathrm{per}} \to z_{\mathrm{per}}}(P(p), K^*(p))\|_\infty =: \mathcal{P}^*(p)$ gives the best possible H_∞ performance plotted over the interval $p \in \Pi = [315, 1100]$. Clearly the parametrization $p \mapsto K^*(p)$ is not practical, and we need approximations of the mapping $K^*(\cdot)$ which can be stored conveniently. This leads to a trade-off between the quantity to be stored and the unavoidable loss of performance. In order to control the loss of performance we adopt the following convention. Fixing $\alpha > 0$, we call a parametrization $p \mapsto K(p)$ acceptable, if

(i) $K(p)$ has structure (4),

(ii) $K(p)$ stabilizes $P(p)$ internally for every p, and

(iii) $\|T_{w_{\mathrm{per}} \to z_{\mathrm{per}}}(P(p), K(p))\|_\infty \leq (1+\alpha)\|T_{w_{\mathrm{per}} \to z_{\mathrm{per}}}(P(p), K^*(p))\|_\infty$ for every p.

To get a scheduling function $K(\cdot)$ which needs as few elements to store (to embed) as possible, we discuss two approaches, which use either interpolation, or switching. For switching we identify subintervals $I = [p_1, p_2] \subset \Pi$ as large as possible on which we can represent $K(p)$ as an affine function without violating criteria (i) - (iii). Then we cover Π with as few as possible of these subintervals $I_1, \ldots, I_N$. The rule to control $P(p)$ is then by switching between these I_i. The way to construct I together with an affine representation $K(p)$ valid on I is given in the next section. A variation which uses interpolation is given in section 8.

Remark. We recall that condition (ii) is only necessary but not sufficient for stability of the switched or interpolated closed loop system. In general it is difficult to prove stability over the parameter domain if nothing is known about the parameter trajectory $p(t)$. Sufficient conditions based on prior bounds $|\dot{p}(t)| \leq \nu$ can be stated, but are generally hard to establish due

to the inherent conservatism. In the same vein, condition (iii) should be understood as a worst case point of view. Namely, unlike in chess where it would be enough to know the best move $K^*(p)$ in any given position p, the situation here is more complicated, because the payoff may be different in different parameter regions, and the seemingly "best" move $K^*(p)$ at p might lead to an unfortunate parameter trajectory in the future. (A similar situation would arise in chess as soon as e.g. winning with more material on the board would count more than winning with few pieces left. The notion of a "best" move would then have to be re-defined).

5.1 LPV model of Magnetic Bearing

For fixed $p_0 \in \Pi$ we wish to find an interval $I(p_0)$ containing p_0 as large as possible such that we have a simple controller parametrization $K(p)$ valid in the sense of (i) – (iii) on $I(p_0)$. In order to construct these intervals, it is helpful to change the parametrization and write $p = p_0(1+\delta)$. The new parameter δ is now centered at 0 and more symmetric, $|\delta| \leq r$, and we wish to have a range of validity r as large as possible. We re-write the open-loop system accordingly. For example, for the third state in (3) we have

$$\begin{aligned} \dot{x}_3 &= \tfrac{-4c_2}{m}x_1 - \tfrac{pJ_\alpha}{J_r}x_4 + \tfrac{2c_1}{m}x_5 \\ &= \tfrac{-4c_2}{m}x_1 - \tfrac{(p_0(1+\delta))J_\alpha}{J_r}x_4 + \tfrac{2c_1}{m}x_5. \end{aligned}$$

Introducing an auxiliary input $w_{\text{rob},1}$ and output $z_{\text{rob},1}$ via $w_{\text{rob},1} = -\frac{J_\alpha}{J_r}\, p_0\, x_4$ and $z_{\text{rob},1} = \delta \cdot w_{\text{rob},1}$, we obtain

$$\dot{x}_3 = \tfrac{-4c_2}{m}x_1 - \tfrac{p_0 J_\alpha}{J_r}x_4 + \tfrac{2c_1}{m}x_5 + z_{\text{rob},1}.$$

Repeating the same thing for all states leads to a standard representation of $P(p)$, $p = p_0(1+\delta)$, as an uncertain system:

$$\begin{bmatrix} \dot{x} \\ z_{\text{rob}_1} \\ y \end{bmatrix} = \begin{bmatrix} A_{\text{sys}}(p_0) & B & B_{1,\text{sys}} & B_{2,\text{sys}} \\ C & 0 & 0 & 0 \\ C_{2,\text{sys}} & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ w_{\text{rob}_1} \\ d \\ u \end{bmatrix} \tag{6}$$

where $z_{\text{rob}_1} \in \mathbb{R}^4$, $w_{\text{rob}_1} \in \mathbb{R}^4$, $w_{\text{rob}_1} = \Delta_1 z_{\text{rob}_1}$ with $\Delta_1 = \delta I_4$ and:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & 0 & \frac{-J_\alpha}{J_r} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{J_\alpha}{J_r} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times p_0.$$

5.2 LPV model of decentralized PID controller

We repeat the same procedure for the controller parametrization (4). We write each of the 8 scheduling functions as $r_i(p) = r_{i0} + \delta \cdot r_{i1}$, and so on until $d'(p) = d'_0 + \delta \cdot d'_1$, with 16 parameters $r_{i0}, \ldots, d'_1$ to be determined for each $I(p_0)$. Notice that $r_{i0} = r_i(p_0)$ depends on the choice of p_0, and similarly for the other scheduling functions. It is helpful to denote this controller as $K(p_0, \delta) = K(p_0) + \delta \cdot dK(p_0)$, even though it still has the form (4).

Having symmetrized the parameter, we get a similar uncertain block $\Delta_2 = \delta \cdot I_8$ with δ repeated 8 times, 4 times for each PID. Altogether this leads to the scheduled controller structure shown in Figure 3.

$$\left[\begin{array}{c} \dot{\xi}_1 \\ \dot{\xi}_2 \\ \hline z_1^k \\ z_2^k \\ z_3^k \\ z_4^k \\ \hline u \end{array}\right] = \left[\begin{array}{cc|cccc|c} 0 & 0 & 1 & 0 & 0 & 0 & r_{i0} \\ 0 & -\tau_0 & 0 & 1 & 1 & 0 & r_{d0} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & r_{i1} \\ 0 & -\tau_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r_{d1} \\ 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 & d_0 \end{array}\right] \left[\begin{array}{c} \xi_1 \\ \xi_2 \\ \hline w_1^k \\ w_2^k \\ w_3^k \\ w_4^k \\ \hline y \end{array}\right], \tag{7}$$

where $\Delta_{21} = \delta \cdot I_4$. We get a similar block for the second PID with primed parameters in (4). This leads to $w_{\text{rob}_2} = \Delta_2 z_{\text{rob}_2}$ where $\Delta_2 = \text{diag}(\Delta_{21}, \Delta_{22}) = \delta I_8$ for K. We then apply a standard procedure to the whole LFT which gives us what is on the right of figure 3. The controller is now independent of δ and gathers the 16 unknown coefficients of the scheduling functions, or rather, the free parameters in (4). The uncertain block is of size $\Delta = \text{diag}(\Delta_1, \Delta_2) = \delta \cdot I_{12}$.

6 ROBUSTNESS INDEX

The constellation on the right of Figure 3 corresponds to the LFT

$$P_{\rm rob}(p_0):\left[\begin{array}{c}\dot{x}\\ \hline z_{\rm rob}\\ y\end{array}\right]=\left[\begin{array}{c|cc}A(p_0) & B_0 & B_2\\ \hline C_0 & D_{00} & D_{02}\\ C_2 & D_{20} & 0\end{array}\right]\left[\begin{array}{c}x\\ \hline w_{\rm rob}\\ u\end{array}\right], \quad (8)$$

$$w_{\rm rob}=\Delta z_{\rm rob},\quad u=\widetilde{K}(p_0)y. \quad (9)$$

The matrix dimensions are $\widetilde{K}(p_0)\in\mathbb{R}^{14\times 14}$, $A(p_0)\in\mathbb{R}^{12\times 12}$, $\Delta\in\mathbb{R}^{12\times 12}$, etc. Notice that $\widetilde{K}(p_0)$ carries the same information as in $K(p_0,\delta)=K(p_0)+\delta\cdot dK(p_0)$, that is, it regroups the 16 unknown coefficients of $K(p_0)$ and $dK(p_0)$ which we want to compute. This

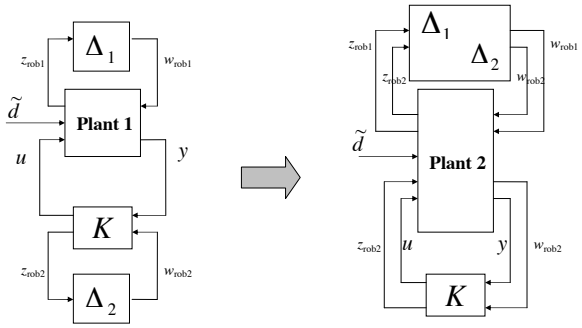

Figure 3: LFT scheme

is now a known situation in robust respectively LPV control. If K in (7) was not structured, we could for instance use mu-tools to design full-order parametric robust controller of size 14×14 which makes the interval of robustness $|\delta|\le r$ as large as possible. Similarly, the LPV procedure of Scherer (Scherer, 2003) would lead to a full order solution. The necessity to satisfy the structural constraint (7) complicates matters, and we have to take recourse to a heuristic method.

We concentrate on stabilizing $K(p,\delta)$ respectively $\widetilde{K}(p)$ on as large an interval $I(p)$ as possible, so we consider (8) without performance channel. The closed-loop system matrix is then

$$A(\widetilde{K},\delta)=A+B_2\widetilde{K}C_2+(B_0+B_2\widetilde{K}D_{20})\Delta\times\ldots$$
$$\times(I-D_{00}\Delta-D_{02}\widetilde{K}D_{20}\Delta)^{-1}(C_0+D_{02}\widetilde{K}C_2) \quad (10)$$

which we re-write as

$$A(\widetilde{K},\delta)=\tilde{A}+\tilde{B}\Delta(I-\tilde{D}\Delta)^{-1}\tilde{C}$$

with $\tilde{A}=A+B_2\widetilde{K}C_2$, etc. Our goal is to guarantee stability of this matrix for as large a range $|\delta|\le r$ as possible. We interprete (10) as the semi-structured complex stability radius $r_{\mathbb{C}}(A+B_2\widetilde{K}C_2)$ of the nominal closed-loop matrix $\tilde{A}=A+B_2\widetilde{K}C_2$. It is well-known that

$$r_C^{-1}(\tilde{A})=\left\|\tilde{C}(sI-\tilde{A})^{-1}\tilde{B}+\tilde{D}\right\|_\infty,$$

so that computing the stability radius of $A+B_2\widetilde{K}C_2$ amounts to computing an H_∞-norm. Altogether, we have

$$r_{\mathbb{C}}^{-1}(\tilde{A})=\|T_{w_{\rm rob}\to z_{\rm rob}}(P_{\rm rob},\widetilde{K})\|_\infty,$$

where $P_{\rm rob}$ is the plant in (8).

7 OPTIMIZATION PROGRAM

For every fixed p_0 we now compute the solution $K_{\rm rob}(p_0)$ of the following mixed H_∞/H_∞ optimization program:

$$\begin{array}{ll}\text{minimize} & \mathcal{R}(K)=\|T_{w_{\rm rob}\to z_{\rm rob}}(P_{\rm rob},\widetilde{K})\|_\infty\\ \text{subject to} & \mathcal{P}(K)=\|T_{w_{\rm per}\to z_{\rm per}}(P(p_0),K)\|_\infty\\ & \qquad\le(1+\beta)\mathcal{P}^*(p_0)\\ & K\text{ has structure (4)}\\ & K\text{ stabilizes internally}\end{array} \quad (11)$$

where $\mathcal{P}^*(p_0)$ is the nominal performance at p_0, that is, $\mathcal{P}^*(p_0)=\|T_{w_{\rm per}\to z_{\rm per}}(P(p_0),K^*(p_0))\|_\infty$. This program presents a trade-off between performance and robustness in the sense of $r_{\mathbb{C}}$. Namely, as we know, the best possible performance at p_0 is obtained by $K^*(p_0)$, which corresponds to choosing $r_{i0}=r_i^*(p_0)$, etc. and $\delta=0$. In (11) we accept a loss of $100\cdot\beta\%$ performance over the nominal value $\mathcal{P}^*(p_0)$ and use this freedom to buy some additional robustness in the sense of $r_{\mathbb{C}}$, hoping that this will lead to a controller $K_{\rm rob}(p_0)$ with as large an interval of validity $I(p_0)$ as possible. Clearly, in order to respect rule (iii), be have to choose $\beta<\alpha$, and in our experiments we use $\beta=\alpha/2$.

The gain scheduling function is now as follows. Let $\nu(p)$ be such that $p\in I(p_\nu)$ and define the control law as $K(p)=K_{\rm rob}(p_{\nu(p)})$. If there are several ν with $p\in I(p_\nu)$, then we use the hysteresis rule. That is, we stay on the interval in which we are, and only jump on a new one when we reach the boundary of the first one. This means that for a value p in the intersection $I(p_\nu)\cap I(p_\mu)$ the actual control law changes depending whether we arrive from the right or from the left.

8 EXPERIMENTAL RESULTS

The measured parameter p varies in $\Pi=[315,\ 1100]\,rad/sec$. We use `hinfstruct` at 51

Algorithm 1 . Algorithm to compute parametrized PID with switching

Parameters: $\alpha > 0, 0 < \beta < \alpha$.

1: Pre-compute approximation of optimal curve $K^*(p)$ using `hinfstruct`.
2: For a sufficient number of parameters p solve mixed H_∞/H_∞ program (11). The solution curve is $K_{\rm rob}(p)$.
3: For every p find interval of validity $I(p)$ of $K_{\rm rob}(p)$ using conditions (i) – (iii).
4: Remove a small portion on each side of $I(p)$ and call the shrunk interval $I^\sharp(p)$.
5: Select minimum number of $I^\sharp(p_\nu)$, $\nu = 1,\dots,N$, covering Π. This means the intervals $I(p_\nu)$ cover Π with some slight overlap.

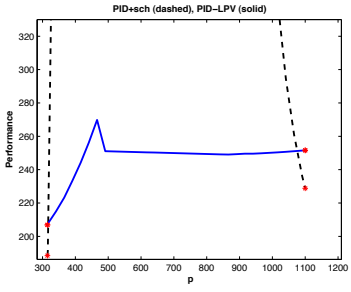

Figure 4: Interpolating linearly between the two controllers $K^*(315)$ and $K^*(1100)$ (dashed line) fails due to a loss of stability, while switching using $K_{\rm rob}(315)$ and $K_{\rm rob}(1100)$ holds the stability (solid line).

equidistant points in Π to compute the optimal decentralized PID controller curve $K^*(p)$.

For each of the 51 values p_0 we use program (11) with $\beta = \alpha/2$ to compute 51 controllers $K_{\rm rob}(p)$, satisfying (i) – (iii) in tandem with their intervals of validity $I(p)$.

First study:

Our first test uses $\alpha = 0.38$ which allows to cover Π with only two intervals $I(315)$ and $I(1100)$. As one can see in Figure 4, the switched controller (continuous line) maintains stability over the entire Π, but interpolating $K^*(315)$ and $K^*(1100)$leads to a loss of stability (broken line). This indicates that interpolation of controllers is more delicate to arrange for than switching.

Second study:

Our second test uses $\alpha = 0.1$, which leads to more realistic results. We seek a switching controller and compute intervals $I(p)$ according to algorithm 1 for 51 equidistant values $p \in \Pi$.

Figure 5 (a) shows the validity region for each of the 51 controllers $K_{\rm rob}(p)$, that is, the values p' for

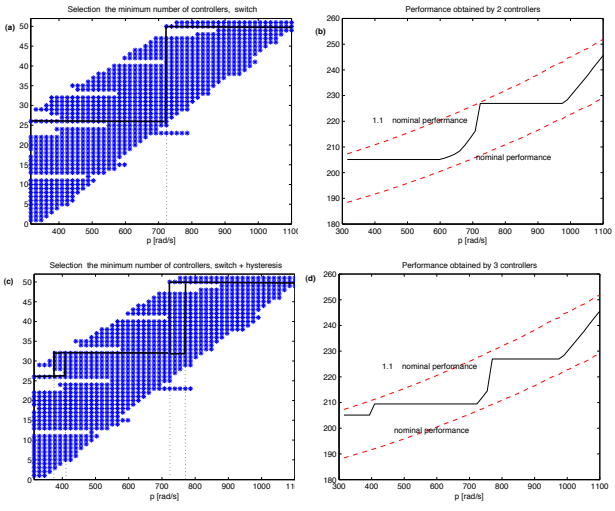

Figure 5: (a) and (c): Validity regions $I(p)$ plotted against p. This allows to read off the intervals needed to cover Π. (b) and (d) the final performance obtained for each case.

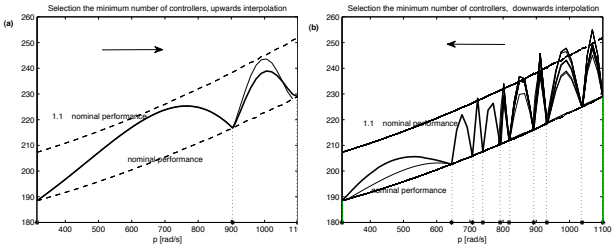

Figure 6: By upwards interpolation, (**a**), we need 3 controllers to cover Π and by downwards interpolation, (**b**), we need 10. Thin lines shows bisection procedure and thick line the final performance.

which controller $K(p)$ satisfies (i) – (iii). For each of the 51 controllers $K_{\rm rob}(p)$ the zone where (i) – (iii) hold corresponds to a horizontal array marked by $*$. As can be seen, by following the thick line, we can find a way covering all the variation of p. This requires only two controllers $K_{\rm rob}(707.5)$ and $K_{\rm rob}(1084.3)$ with switching at $p = 723.2\,rad/s$ (controllers number 26 and 50).

If we insist on a non-negligible overlap between the intervals $I(p)$, using the $I^\sharp(p)$ as in the algorithm, then Figure 5 (c) shows that we need three controllers $K_{\rm rob}(707.5)$, $K_{\rm rob}(801.7)$ and $K_{\rm rob}(1084.3)$, to cover Π (controllers number 26, 32 and 50). The same Figures 5 (b) and (d) show the performance obtained in each case.

Third study:

Our third test still uses $\alpha = 0.1$, but now we interpolate optimal controllers $K^*(p)$. Start at the left end

$p_1 = 315$. Having found p_k, we examine for $p > p_k$ the closed-loop performance curve obtained by the controller $K_{\text{int}}(p_k, p)$ interpolating between $K^*(p_k)$ and $K^*(p)$. As p increases, this curve eventually hits the upper limit curve $(1+\alpha)\mathcal{P}(K^*(p'))$ at some p' between p_k and p. We put $p_{k+1} = p$ and continue until the right end point is reached. A similar procedure starts at the right end point and moves downward.

Figure 6 gives the results of this algorithm in our study. Upwards interpolation needs 3 controllers $K^*(p)$ at $p = 315, 903.75, 1100$, while downwards interpolation requires more, namely $K^*(p)$ at $p = 1100, 1037.7, 931.7, 892.2, 817.9, 790.9, 707.5, 644.65, 315$.

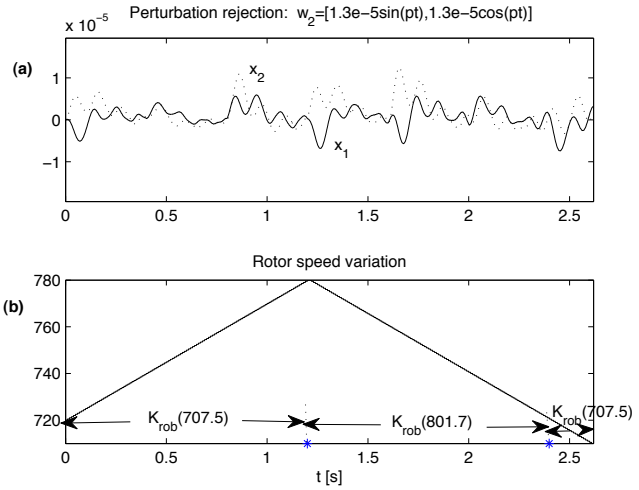

Figure 7: Simulation in closed loop. The scheduled parameter increases within 1.2 sec from 720 to 780, and decreases back to 710 within another 1.5 sec. Three controllers $K_{\text{rob}}(p(t))$ are called for. Upper image shows unbalance compensation x_1, x_2 for simulated $w_2(t) = (1.3e-5\sin p(t), 1.3e-5\cos p(t))$. (For x_1, x_2, w_2 see section 2).

9 CONCLUSION

Several methods to compute a parameter varying decentralized PID for a magnetic bearing device were compared. Performance was measured in the H_∞ norm, and the curve $K^*(p)$ of optimal H_∞-controllers was taken as a reference to assess the performance of the different parameterizations $K(p)$. If parameterizations $K(p)$ with a maximum loss of 10% over $K^*(p)$ were allowed, switching between piecewise affine controllers on subintervals was found to perform best, but needs solving a mixed H_∞/H_∞ synthesis program. Interpolation based on computing various $K^*(p)$ was an interesting alternative, even though it was observed that interpolation seems to have a stronger tendency to lose stability and important dependence at the beginning point. While the switching technique carries over to 2D parameter sets, there is no obvious way to extend interpolation into two dimensions.

REFERENCES

Apkarian, P. and Gahinet, P. (1996). A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. In *IEEE Transactions on Control Systems Technology, vol. 40, No. 5, pp. 835-864*.

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. In *Automatica, vol. 31, No. 9, pp. 1251-1261*.

D. Hinrichsen, A. P. (1986a). Stability radii of linear systems. In *Systems and Control Letters, vol. 7, pp. 1-10*.

D. Hinrichsen, A. P. (1986b). Stability radius for structured perturbations and the algebraic riccati equation. In *Systems and Control Letters, vol. 8, pp. 105-113*.

Lanzon, A. and Tsiotras, P. (2005). A combined application of $\mathcal{H}_\infty$ loop shaping and μ- synthesis to control high-speed flywheels. In *IEEE Transactions on Control Systems Technology, vol. 13, No. 5*.

Lawrence, C., Tits, A. L., and Dooren, P. (2000). A fast algorithm for the computation of an upper bound on the μ-norm. In *Automatica, vol. 36, pp. 449-456*.

M. Karow, E. Kokiopoulou, D. K. (2010). On the computation of structured singular values and pseudospectra. In *Systems and Control Letters, vol. 59, No. 2, pp. 122-129*.

Mohamed, A. and Busch-Vishniac, I. (1995). Imbalance compensation and automation balancing in magnetic bearing systems using the q-parametrization theory. In *IEEE Transactions on Control Systems Technology, vol. 3, No. 2*.

Packard, A. (1994). Gain scheduling via linear fractional transformations. In *Systems and Control Letters, vol. 22, pp. 79-92*.

R.D. Smith, W. W. (1995). Nonlinear control of a rigid rotor magnetic bearing system: modeling and simulation with full state feedback. In *IEEE Transactions on Magnetics, vol. 31, no. 2, pp. 973 - 980*.

Scherer, C. (2003). Higher-order relaxations for robust lmi problems with verification for exactness. In *Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 4652 - 4657*.

Tsiotras, P. and Mason, S. (1996). Self-sheduled $\mathcal{H}_\infty$ controllers for magnetic bearings. In *Proc. Int. Mechanical Engineering Congr. Exposition, Atlanta, GA, pp. 151-158*.