

GAIN-SCHEDULED CONTROL FOR SUBSTRUCTURE PROPERTIES IN AMBs

Stephen J. Fedigan,¹ Carl R. Knospe,² Ronald D. Williams³

ABSTRACT

In AMB systems, machine performance can be degraded by the dynamics of the substructure (i.e. the object to which the sensors and actuators are mounted). These dynamics, unlike those of the rotor, actuators, and sensors are not completely known at the time of controller design. This paper introduces a novel approach called *a posteriori* gain scheduling for addressing this problem and evaluates the suitability of LPV and D-K iteration design methods by applying them to a benchmark problem. An LPV design method modified for time-invariant parameters exhibits performance close to H_∞ optimal designs, and is modified to account for parameter estimation errors. The resulting robust, gain-scheduled controller exhibits performance close to mixed- μ controllers designed using D,G-K iteration.

1. INTRODUCTION

In many active magnetic bearing (AMB) systems, the dynamics of the substructure (i.e. the object to which the actuators and sensors are mounted) can significantly impact machine performance (Lantto, Väänänen, and Antila, 1996). However, unlike the rotor, actuator, and sensor dynamics, these substructure properties, which include both governing equations and physical parameters, may not be completely known at the time of controller design.

One way to approach this problem might be to develop an accurate model of the substructure by *in situ* testing, and synthesize a controller off-line using this identified model. This is practical if the AMB system is either a one-of-a-kind installation or is manufactured in limited quantities. However, such an approach becomes prohibitively expensive if the AMB system is in mass production as it involves fielding trained personnel to re-tune (or even re-design) the controller at each site. An alternative to this labor intensive approach involves treating these unknown dynamics as either

¹Texas Instruments, Inc., M/S 8368, P.O. Box 655303, Dallas, TX 75265, USA, tel.: (972) 997-6147, fax: (972) 997-5693, e-mail: fedigan@ti.com.

²Department of Mechanical, Aerospace, and Nuclear Engineering, Mechanical Engineering Bldg., University of Virginia, Charlottesville, VA 22903-2448, USA, tel.: (804) 982-2603, fax: (804) 982-2246, e-mail: crk4y@virginia.edu.

³Department of Electrical Engineering, Thornton Hall, University of Virginia, Charlottesville, VA 22903-2448, USA, tel.: (804)924-7960, fax: (804)924-8818, e-mail: rdw@virginia.edu.

parametric or dynamic uncertainties. This is a valid approach if the uncertainties are either small in size or if the performance requirements are not too stringent. Otherwise, this approach may be excessively conservative. Still another approach is to use on-line adaptive control. This approach is basically turn-key, provided the AMB system can withstand a long transient period as the adaptive controller learns and controls the unknown substructure dynamics. While an adaptive controller may be able to optimize the closed loop system with respect to some performance measure, there are often no guarantees against system uncertainties, and in some cases, performance may degrade rapidly as the actual plant deviates from the model assumed in the controller design.

In this paper, a new approach which will be called *a posteriori* gain-scheduling is investigated for solving this problem. In brief, the method involves synthesizing a controller which is gain-scheduled on unknown substructure properties, estimating these properties, and entering them into the gain-scheduled controller. For AMB systems with wide variations in substructure properties, this method provides a potential way to meet demanding performance specifications with a low cost controller. By employing recently-developed implicit gain-scheduling methods (Apkarian and Gahinet, 1995; Packard, 1994), system performance can be guaranteed and the hazards of traditional gain-scheduling (Shamma and Athans, 1992) can be avoided without the need to re-design the controller for each new substructure. For a mass-production application, not only do these methods avoid an expensive re-design, they also keep down the cost of the embedded microprocessor control system because they generate compact descriptions of gain-scheduled controllers that have low storage requirements.

2. SCOPE OF PAPER

In this paper, only the controller design aspect of *a posteriori* gain-scheduling will be addressed, leaving the system identification element as a topic for future publications. Section 3 will introduce a benchmark problem that will be used for evaluating different gain-scheduled controllers. Section 4 will review a gain-scheduled synthesis method based on D-K (D,G-K) iteration, which will hereafter will referred to D-K (D,G-K) gain-scheduled synthesis, and Section 5 will explain how LPV (Linear Parameter Varying) design methods can be used to develop controllers scheduled on time-invariant parameters. Section 6 will place the performance of these two types of gain-scheduled controllers into context by comparing them against optimal H_∞ point designs and controllers that are robust over the same range of parameter values. Section 7 will incorporate robustness against parameter estimation errors into the benchmark design via the methods of (Apkarian and Adams, 1997), and will present the performance of the resulting robust LPV controllers.

3. BENCHMARK PROBLEM

A simple rotor-bearing system model connected in parallel with a substructure

stiffness is written as

$$k_s = k_{s_0}(1 + w_{k_s}\delta_{k_s}) \quad w_{k_s} = 0.90 \quad \delta_{k_s} \in [-1, 1] \quad (1)$$

where k_{s_0} represents its nominal value, w_{k_s} its percentage range of variation, and δ_{k_s} its variation from nominal. The plant is written as a linear fractional transformation (LFT) on the nominal plant and the gain-scheduling parameter. As is standard practice in the μ -synthesis framework (Balas et al., 1995), the additional I/O pair that is created is treated as an additional disturbance input w_k and error output z_k . It should be noted that when this loop is closed for a specific value of δ_{k_s} , only a single entry in the plant's A matrix is affected. To place a penalty on control effort, an additional I/O pair is introduced. The new input w_n is injected right before the controller and the new output z_n is taken right after the controller. By appropriately weighting this new performance output, an upper limit on controller gain may be established.

4. D-K (D,G-K) GAIN-SCHEDULED SYNTHESIS

In D-K gain-scheduled synthesis (Balas, Packard, Becker, 1992; Helmersson, 1995; Lu and Balas, 1995), an additional copy of the δ_{k_s} parameter is supplied to the controller as shown in Figure 2. In order to put the problem into the standard form required by μ -synthesis (Balas et al., 1995), a repeated "uncertainty" block is formed and the additional copy of δ_{k_s} is fed through an augmented plant (original plant + wires) to the controller. At this point, the arrangement looks like a standard μ -synthesis problem with a repeated uncertainty block. The controller design process is identical to the standard D-K iteration procedure with the exception that the initial controller design is an LPV gain-scheduled controller. This modification (Balas, Packard, and Becker, 1992; Helmersson, 1995) is necessary for promoting the development of gain-scheduled controllers, since a standard H_∞ controller will always sever the feedthru connections for the unscaled system to optimize disturbance rejection and therefore will progress towards a robust controller design. After the full D-scales are appended to the system, the coupling provided by these scales will prevent the H_∞ controller from breaking the connection and designing a robust controller. To improve performance, a sequence of D-K iterations can be followed up with a few D,G-K iterations (Young, 1994), which treat the parameter as a real rather than a complex time-invariant quantity.

5. LPV CONTROLLER SYNTHESIS FOR TIME-INVARIANT PARAMETERS

In LPV gain-scheduled synthesis, the benchmark problem is cast as a linear objective minimization (MINCX) (Boyd et al., 1994) subject to a system of LMI (Linear Matrix Inequality) constraints which are continuous in the gain-scheduling parameter. Since we desire a controller which is a direct function of the gain-scheduling parameter, the "basic" characterization (Apkarian and Adams, 1997) is preferred over the

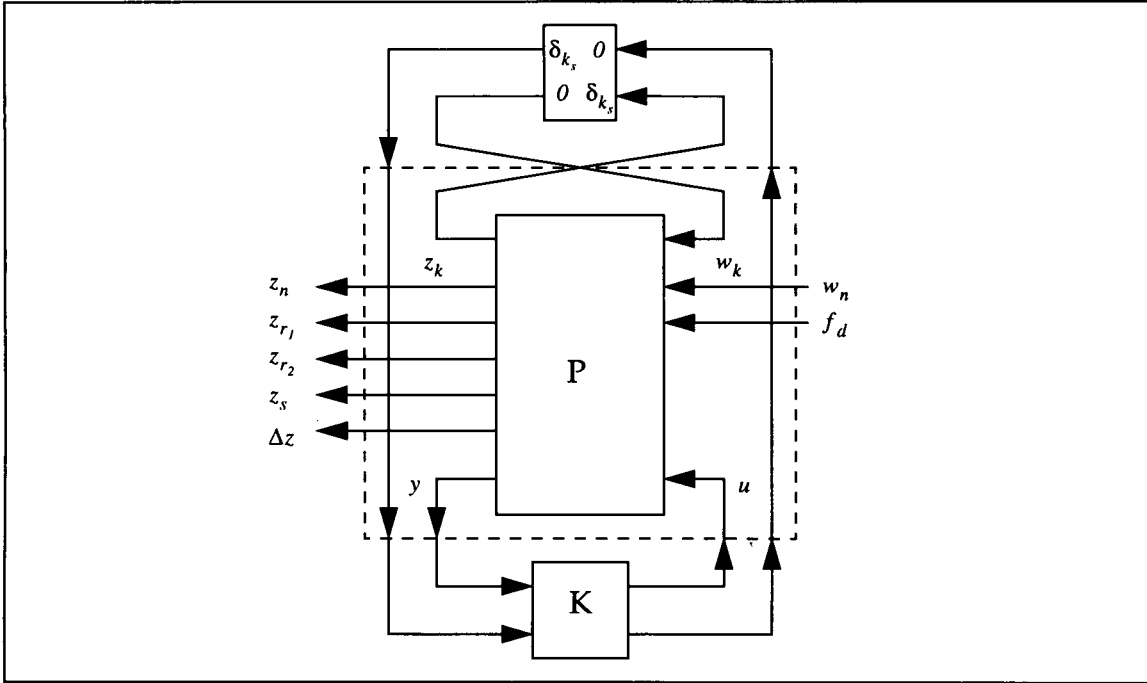


Figure 2: Augmented plant for D-K gain-scheduled synthesis.

“projected” characterization (Apkarian and Adams, 1997). Specifically, for the benchmark problem, the variable γ (induced L_2 norm) is minimized subject to the constraints

$$\Psi(\delta) = \begin{bmatrix} X(\delta)A(\delta) + \hat{B}_k(\delta)C_2 + (\star) & \star & \star & \star \\ \hat{A}_k^T(\delta) & A(\delta)Y(\delta) + B_2\hat{C}_k(\delta) + (\star) & \star & \star \\ [X(\delta)B_1 + \hat{B}_k(\delta)D_{21}]^T & [B_1 + B_2D_k(\delta)D_{21}]^T & -\gamma I & \star \\ C_1 + D_{12}D_k(\delta)C_2 & C_1Y(\delta) + D_{12}\hat{C}_k(\delta) & D_{11} + D_{12}D_k(\delta)D_{21} & -\gamma I \end{bmatrix} < 0 \quad (2)$$

$$\begin{bmatrix} X(\delta) & I \\ I & Y(\delta) \end{bmatrix} > 0 \quad (3)$$

over the storage variables $\{X(\delta), Y(\delta), \gamma\}$ and the transformed controller variables $\{\hat{A}_k^T(\delta), \hat{B}_k(\delta), \hat{C}_k(\delta), D_k(\delta)\}$ where $\delta \in [-1, 1]$ is the normalized substructure stiffness and where the symbol \star designates symmetric matrix completion. In equation (2), the plant has been divided into two sets of inputs and outputs with state-space representation

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\delta) & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \quad (4)$$

where plant dimensions are summarized by $A \in R^{n \times n}$, $D_{11} \in R^{n_z \times n_w}$, and $D_{22} \in R^{n_y \times n_u}$. The signals w and u represent disturbance and control inputs respectively, and the signals z and y represent controlled and measurement outputs respectively. In (4), D_{22} is assumed to be zero without loss of generality. After the optimization is completed, the transformed controller variables are converted into the original controller variables in the two step procedure (Gahinet, 1996) which involves a series of elementary matrix manipulations.

To turn this into a problem which can be submitted to an LMI solver, the functional dependence of the optimization variables on the gain-scheduling parameter must be specified (Becker, 1996; Wu et al., 1996). While no procedure yet exists for determining the functional dependence which leads to the best performance, copying the substructure's affine parameter dependence (Wu et al., 1996) is common practice and normally yields good performance. Following this rule of thumb, we shall choose affine parameter dependence for our optimization variables:

$$\begin{aligned} X(\delta) &= X_0 + \delta X_1 & Y(\delta) &= Y_0 + \delta Y_1 \\ \hat{A}_k(\delta) &= \hat{A}_{k_0} + \delta \hat{A}_{k_1} & \hat{B}_k(\delta) &= \hat{B}_{k_0} + \delta \hat{B}_{k_1} \\ \hat{C}_k(\delta) &= \hat{C}_{k_0} + \delta \hat{C}_{k_1} & D_k(\delta) &= D_{k_0} + \delta D_{k_1} \end{aligned} \quad (5)$$

In their current form, the LMI constraints (2) and (3) must be met for every value in the range of the gain-scheduling parameter. Constraint (3) presents no difficulty since it is affine in the parameter and its satisfaction is assured by meeting the constraint at the extreme values of δ . Although removing the parameter dependence from $\{X, Y\}$ is an obvious way to make (2) tractable, the resulting controller would guard against arbitrarily-fast parameter variation rates. This would be especially conservative for this benchmark problem, considering the parameter is actually time-invariant.

A second way of "discretizing" (2) involves gridding (Becker, 1996; Wu et al., 1996) the parameter space and enforcing (2) on the grid points. This is a necessary but not sufficient condition for meeting (2) throughout the parameter space and offers no performance guarantees when the parameter value falls between grid points. However, this can provide a tight lower bound on performance if design grid is sufficiently dense, and when the design is completed, (2) can be checked on a more refined grid. It is practical for a small number of parameters (perhaps up to 3) but it rapidly becomes impractical since the number of LMI constraints grows geometrically with the dimension of the parameter space.

This dimensional explosion in LMI constraints can be avoided when an LMI condition has general polynomial dependence. In this case, it is sufficient to meet (2) at its extreme points, provided that a multi-convexity condition (Gahinet, Apkarian, and Chilali, 1996) is satisfied throughout the range of the parameter. For (2), which exhibits a quadratic parameter dependence, this amounts to adding the following parameter-independent LMIs:

$$X_1 A_1 + (\star) \geq 0 \quad A_1 Y_1 + (\star) \geq 0 \quad (6)$$

6. SYNTHESIS RESULTS

Figure 3 graphs the performance level γ vs. δ of various gain-scheduled, robust, and optimal controllers synthesized for the benchmark problem. The H_∞ point designs deliver optimal performance and are included to establish a baseline. The robust controllers are another convenient point of reference since they establish an upper bound on controller performance. The CX-ROB robust controller, which is synthesized by D-K iteration using commercially available Matlab software (Balas et al., 1995), treats δ as an uncertain, time-invariant, complex quantity of unit magnitude. The MX-ROB robust controller, which is synthesized by D,G-K iteration (an implementation of mixed μ -synthesis) (Young, 1994), is less conservative as it treats δ as a real rather than complex quantity. Between these points of reference, the performance levels of four gain-scheduled controllers, labelled LPV-LTV, LPV-LTI, CX-GS, and MX-GS, are plotted. The first two are designed using LPV gain-scheduling methods. The LPV-LTV controller sacrifices performance in favor of lower design complexity by using parameter-independent $\{x, y\}$ but still retains linear parameter-dependent controller variables. The LPV-LTI controller is synthesized with both linear parameter-dependent storage functions and controller variables, using a 20 point design grid, and the LMIs are verified over a 200 point design grid. The second two, CX-GS and MX-GS, are based respectively on D-K and D,G-K iteration using the feedthru approach described in Section 4. Both methods use an LPV-LTV gain-scheduled controller as an iteration seed and attempt to improve the performance of this initial controller.

The performance of the LPV-LTI controller is particularly impressive and is nearly on par with the H_∞ optimal controller at the negative parameter value. The controller's transfer function, plotted in Figure 4, is also physically appealing; it is essentially a proportional-derivative (PD) controller with roll-off, with one notch at the frequency of the rotor's bending mode and another notch which roughly tracks the substructure's resonance frequency as the stiffness parameter travels throughout its range of values.

In contrast, the LPV-LTV controller exhibits worse performance than even the CX-ROB controller. This design's performance would probably compare more favorably against other design methods if the plant had rapidly evolving instead of time-invariant parameters. However, it shows rather dramatically the importance of including (if it is available) *a priori* parameter rate-information into the design procedure. The LPV-LTV controller also indicates how much D-K (D,G-K) iteration improves upon the performance of the initial controller seed. In the case of D-K-gain-scheduled synthesis, the controller performance lands very close to the D-K robust controller. The controller is a weak function of the parameter value, and its transfer function winds up looking very much like the robust controller's transfer function. This is a surprising result, considering that the gain-scheduled controller really "contains" the robust controller as a special case, and one would expect the controller to take advantage of its "knowledge" of the parameter.

Along these lines, the D,G-K gain-scheduled controller performance is even more surprising, since it cannot even match the performance of the D,G-K robust controller. We believe this in part resulted from the synthesis procedure, which requires a very accurate fit of the full D,G-scales associated with the gain-scheduling

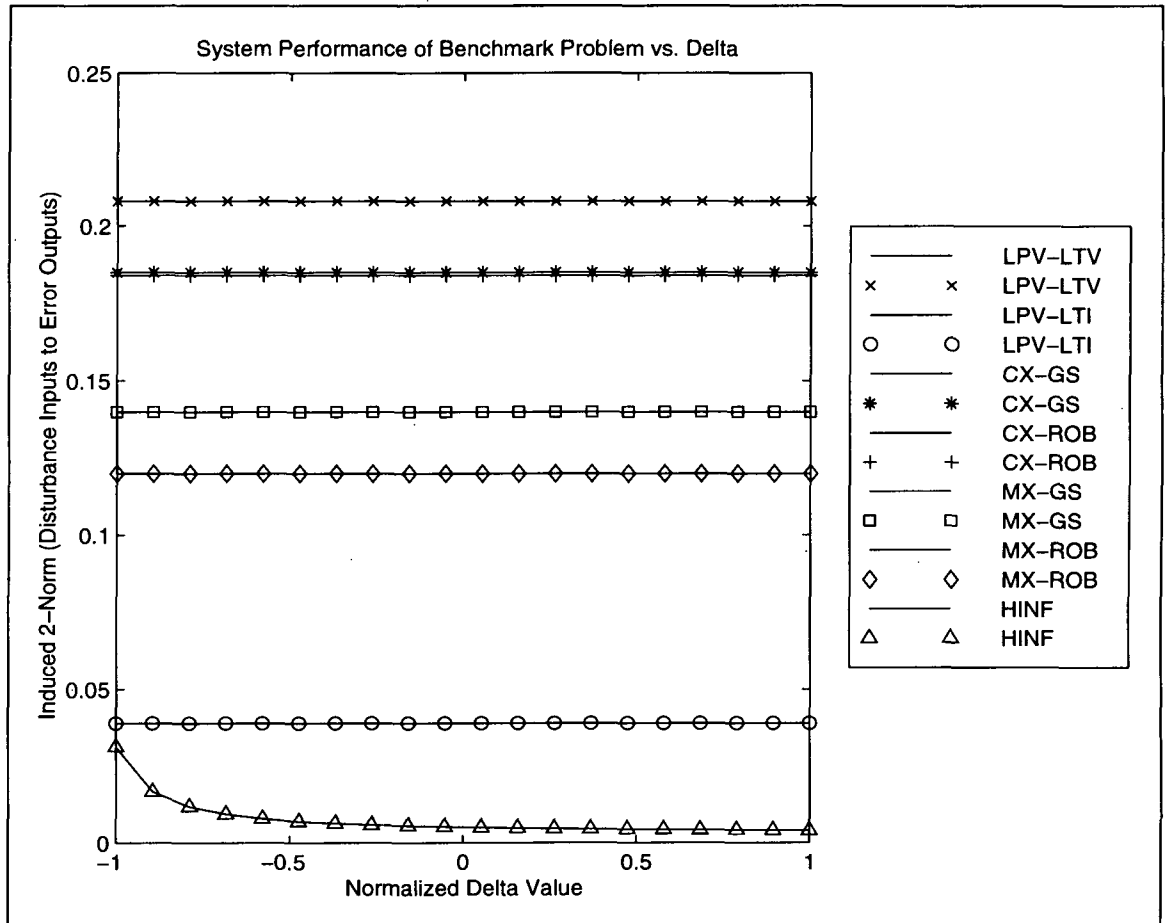


Figure 3: Global performance for various robust and gain scheduled controllers, including the LTI-LPV controller, designed for time-invariant parameters.

“uncertainty” block in order to capture the real parameter’s phase information. If the scaling data is fitted with a high order transfer matrix, the controller order becomes inordinately high (>100 for a 6th order plant), which can lead to numerical problems. On the other hand, if the transfer matrix is not of sufficient order, the scaled system’s norm will not accurately represent the mixed- μ upper bound, and mixed- μ may not decrease monotonically as the iteration proceeds.

7. ROBUST GAIN-SCHEDULED SYNTHESIS

In a typical gain-scheduling scenario, the scheduling parameters are measured explicitly and are accessed directly by the controller. In the *a posteriori* gain-scheduling paradigm, the scheduling parameters are *estimated* using data available from existing control system sensors and actuators. As a result, the parameter values will be subject to a non-negligible estimation error. The current synthesis procedure does not guarantee performance in the presence of these estimation errors, and it is conceivable

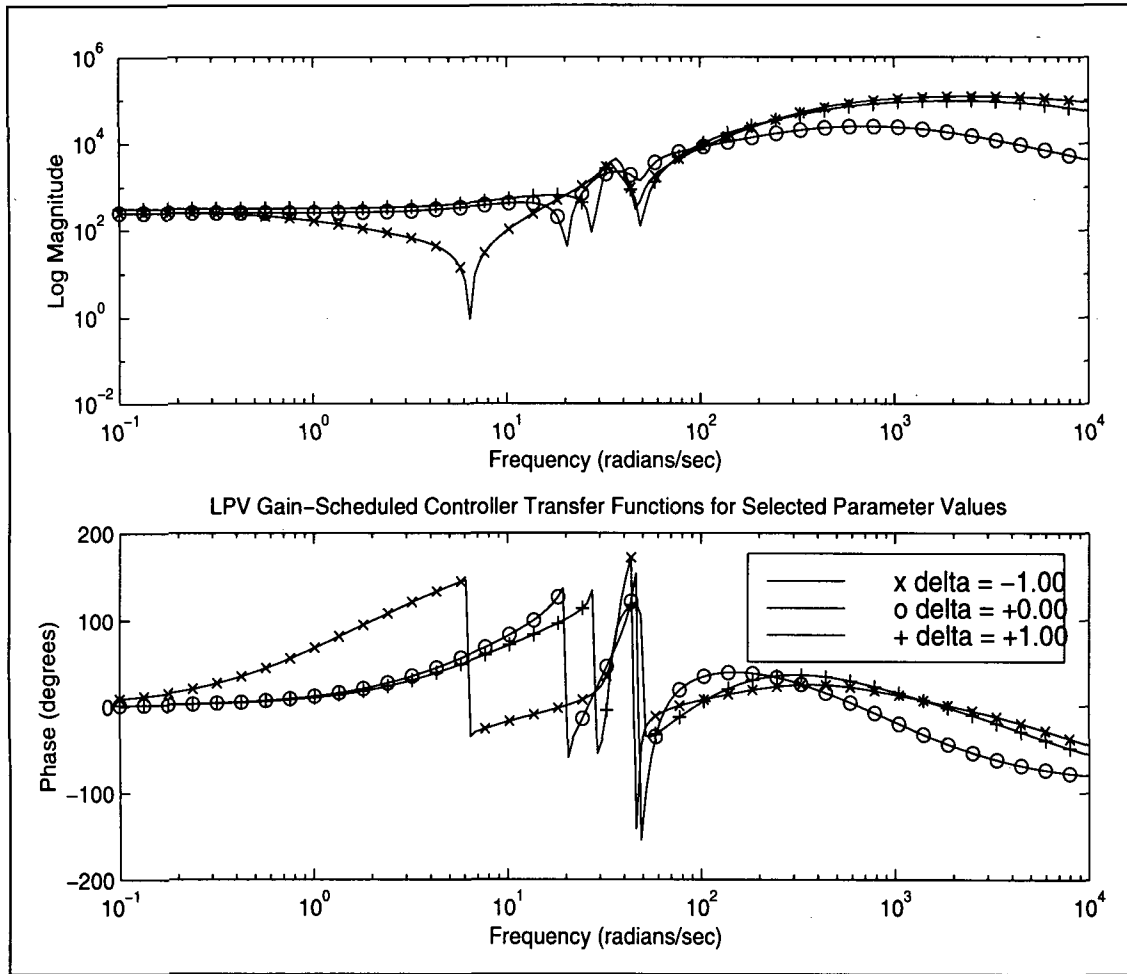


Figure 4: LPV-LTI gain scheduled controller family.

that even a small error could de-stabilize the system. Fortunately, it is not hard to include such robustness into the design procedure. As an illustration, suppose that the estimation error in the benchmark problem is additive, i.e.

$$k_s = \hat{k}_s + 0.05k_{s_0}\delta_e \quad (7)$$

where $\delta_e \in [-1, 1]$. This additive error is weighted such that a nominal stiffness estimate will yield an 5% error. However, this means that lower estimates will have higher percentage errors and higher estimates lower percentage errors. To avoid unreasonably large percentage errors, the stiffness in this next example will only have a 70% range of variation, which will produce an error of 17% at the low end and 3% at the high end. Now that the error level has been established, δ_e is pulled out of the plant through a feedback interconnection, and an additional performance I/O pair is created. As is standard practice in D-K iteration, the performance outputs are scaled by $1/\gamma_1$, and, if we can design a controller such that $\gamma < 1$ for the scaled system, the target performance

γ_i of the original system will be achieved.

Since the performance I/O is now structured, it is advisable to add scales to the system to reduce conservatism. The scales which will be added are basically constant D-scales, which commute with a complex time-varying error. While this is certainly a conservative treatment of the error, the constant scales are incorporated easily into the synthesis procedure and do not add to the order of the controller. As in D-K iteration, the scales must be non-singular and commute with the Δ -blocks. With the scales s and s^{-1} pre- and post-multiplying the plant, the new basic characterization (Apkarian and Adams, 1997) becomes

$$\begin{bmatrix} X(\delta)A(\delta) + \hat{B}_k(\delta)C_2 + (\star) & \star & \star & \star \\ \hat{A}_k^T(\delta) & A(\delta)Y(\delta) + B_2\hat{C}_k(\delta) + (\star) & \star & \star \\ \Sigma(\delta)[X(\delta)B_1 + \hat{B}_k(\delta)D_{21}]^T & \Sigma(\delta)[B_1 + B_2D_k(\delta)D_{21}]^T & -\gamma\Sigma(\delta) & \star \\ C_1 + D_{12}D_k(\delta)C_2 & C_1Y(\delta) + D_{12}\hat{C}_k(\delta) & [D_{11} + D_{12}D_k(\delta)D_{21}]\Sigma(\delta) & -\gamma\Sigma(\delta) \end{bmatrix} < 0 \quad (8)$$

$$\begin{bmatrix} X(\delta) & I \\ I & Y(\delta) \end{bmatrix} > 0 \quad (9)$$

where the scaling $\Sigma := s^{-1}$ is added to the list of optimization variables. Using the method of Apkarian and Adams (Apkarian and Adams, 1997), this non-convex problem is solved by using a two step iteration procedure which is very much like D-K iteration. To start the iteration, Σ is initialized with the identity matrix. In the first step, Σ is fixed with the value from the previous step, and the optimization becomes a linear objective minimization (MINCX) in the remaining variables. In the second step, the variables $\{X, \hat{B}_k, D_k\}$ are fixed with the values from the previous step, and the optimization becomes a generalized eigenvalue (GEVP) problem in the remaining variables.

Performance results for controllers with estimation errors are summarized in the table below:

TABLE 2: LPV AND ROBUST CONTROLLER GLOBAL PERFORMANCE

| # | Controller Design Assumptions | γ |
|---|---|----------|
| 1 | LPV G.S. with multi-convexity, error & structured disturbances with linear parameter-dependent scales | 0.038 |
| 2 | LPV G.S. with gridding, error & structured disturbances with linear parameter-dependent scales | 0.028 |
| 3 | LPV G.S. with gridding, error & structured disturbances with independent scaling variables at each grid point | 0.028 |
| 4 | D-K robust with constant D scales | 0.024 |
| 5 | D-K robust with dynamic D scales | 0.023 |
| 6 | D,G-K robust with dynamic D,G scales | 0.020 |

A number of cases have been considered to gain insight into how various assumptions affect controller performance. In case 1, we applied scales with linear parameter-dependence and required multi-convexity, which avoided the gridding by allowing us to impose constraints at the extreme parameter values. In case 2, we gridded over the parameter's range and used linear parameter-dependent scales. This significantly improved performance over case 1. To assess the cost of linear parameter dependence in the scales, we synthesized a controller which had independent scales at each grid point. Surprisingly, this did not improve the performance significantly.

In case 4, we examined the cost of gain-scheduling by designing a series of controllers robust with respect to the estimation error for various stiffness values, using D-K iteration with constant D scales. The number listed is the worst case performance over all the robust controllers spanning the parameter range. Not surprisingly, the performance is the worst at the low end of the parameter value, which suffers from the largest percentage estimation error. It is encouraging to see that the best robust gain-scheduled controller had very similar performance to that of the worst robust controller.

To see how much controller performance would be improved by treating the estimation error as a complex, time-invariant parameter, we synthesized several robust controllers using D-K iteration with dynamic D-scales at several points over the range of the parameter estimate and took the worst case performance. This did not significantly improve the performance over D-K iteration with constant D scales (case 4). We repeated this exercise by synthesizing several robust controllers using D,G-K iteration with dynamic D and G scales, which treat the parameter as a real, time-invariant quantity. For this problem, it is clear that the dynamic scales used in cases 5 and 6 do not offer significant advantages for this problem, and therefore a gain-scheduled controller synthesized with dynamic scales would not be significantly better than the constant scales employed in case 3.

8. CONCLUSIONS

In this paper, the *a posteriori* gain-scheduling concept is investigated for addressing unknown substructure properties in AMB systems. Rate-bounded LPV controllers with zero parameter-variation rates demonstrate performance close to that of H_∞ optimal point designs. These synthesis methods can be extended to guarantee performance in the face of parameter estimation and/or measurement errors. The performance of these robust, gain-scheduled controllers comes close to mixed- μ synthesis point designs.

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge many helpful discussions with Dr. Pierre Apkarian and the donation of mixed- μ synthesis software by Dr. Peter Young.

REFERENCES

- P. Apkarian and R. Adams. "Advanced Gain-Scheduling Techniques for Uncertain Systems." *Proceedings of the 1997 American Control Conference (ACC)*, Albuquerque, New Mexico, June 1997.
- P. Apkarian and P. Gahinet. "A Convex Characterization of Gain-Scheduled H_∞ Controllers." *IEEE Transactions on Automatic Control*, 40(5): 853-864, May 1995.
- G. Balas, J. Doyle, K. Glover, A. Packard, and R. Smith. *μ -Analysis and Synthesis Toolbox*. The MathWorks, Inc., Natick, Massachusetts and MUSYN, Inc., Minneapolis, Minnesota, 1995.
- G. Balas, A. Packard, and G. Becker. "Robust, Gain-Scheduled Autopilots for Missiles." Naval Weapons Center Final Report, N60530-91-C-0218, MUSYN, Inc., September 1992.
- G. Becker. "Additional Results on Parameter-Dependent Controllers for LPV Systems." *Proceedings of the 1996 International Federation of Automatic Control (IFAC)*, 13th Triennial World Congress, pages 351-356, San Francisco, California, June-July 1996.
- S. Boyd, L. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics, Society of Industrial and Applied Mathematics, Philadelphia, 1994.
- P. Gahinet. "Explicit Controller Formulas for LMI-based H_∞ Synthesis." *Automatica*, 32(7): 1007-1014, July 1996.
- P. Gahinet, P. Apkarian, and M. Chilali. "Affine Parameter-Dependent Lyapunov Functions and Real Parametric Uncertainty." *IEEE Transactions on Automatic Control*, 41(3): 436-442, March 1996.
- A. Helmersson. "Methods for Robust Gain Scheduling." Ph.D. Dissertation, Linköping University, 1995.
- E. Lantto, J. Väänänen, and M. Antila. "Effect of Foundation Stiffness on Active Magnetic Bearing Suspension." *Proceedings of the Fifth International Symposium on Magnetic Bearings*, pages 37-42, Kenazawa, Japan, August, 1996.
- W. Lu and G. Balas. "Robust, Gain-Scheduled Control Design of a Lightly Damped Plant." *Proceedings of the 34th Conference on Decision and Control (CDC)*, New Orleans, Louisiana, December 1995.
- A. Packard. "Gain Scheduling via Linear Fractional Transformations." *Systems & Control Letters*, 22: 79-92, 1994.
- J. Shamma and M. Athans. "Gain Scheduling: Potential Hazards and Possible Remedies," *IEEE Control Systems Magazine*, 12(3): 101-7, March 1992.
- F. Wu, X. Yang, A. Packard, and G. Becker. "Induced L_2 -Norm Control for LPV System with Bounded Parameter Variation Rates." *Proceedings of the 1995 American Control Conference (ACC)*, pages 2379-2383, Seattle, Washington, June 1996.
- P. Young. "Controller Design with Mixed Uncertainties." *Proceedings of the 1994 American Control Conference (ACC)*, pages 2333-2337, Baltimore, Maryland, June 1994.