

Optimal Structure Design using Branch and Bound

Jaime D Sipila*

Mechanical and Aerospace Engineering
 University of California, Los Angeles

Robert M'Closkey†

Andy Packard‡
 Mechanical Engineering
 University of California, Berkeley

Abstract

A branch and bound algorithm is used to determine optimal spring rates and damping coefficients in flexible structures although the formalism is easily adaptable to designing passive control systems. The goal is to minimize the \mathcal{H}_∞ norm from disturbance forces to displacements and velocities of selected degrees of freedom. The success of the branch and bound algorithm largely depends upon the quality of upper and lower bounds for the objective function. A comparison on two design problems is made between a tight, but computationally expensive, lower bound, and a simple, computationally cheap, lower bound. Our conclusion is that the extra investment in time to compute the tight bound is well worth the effort and leads to a dramatic savings in overall computation time.

1 Introduction

This paper considers the computational benefits of several lower bounds in a branch and bound algorithm applied to the design of a multi-degree-of-freedom structure. The design task is formulated as the minimization of the \mathcal{H}_∞ norm of an LFT over a parameter set,

$$\Phi_{\min}(\mathbf{B}_\Delta) := \min_{\Delta \in \mathbf{B}_\Delta} \|F_L(M, \Delta)\|_\infty$$

$$= \min_{\Delta \in \mathbf{B}_\Delta} \left\| \begin{array}{c} z \\ \left[\begin{array}{c} M \\ \Delta \end{array} \right] \begin{array}{c} w \\ u \end{array} \end{array} \right\|_\infty \quad (1)$$

where M represents the linear system dynamics, $\Delta = \text{diag}\{\delta_1 I_{r_1}, \delta_2 I_{r_2}, \dots, \delta_s I_{r_s}\} \in \mathbb{R}^{m \times m}$ is a diagonal matrix of design parameters, and $\mathbf{B}_\Delta = \{\Delta \in \Delta : \bar{\sigma}(\Delta) \leq 1\}$ is the unit ball of normalized parameters (after appropriate loop shifting and y/u -channel scaling). The system is partitioned

$$\text{as } M = \begin{bmatrix} M_{zw} & M_{zu} \\ M_{yw} & M_{yu} \end{bmatrix}.$$

This problem requires global optimization. One of the more popular and promising methods is the branch and bound technique, which we apply to solve (1). We assume that all $\Delta \in \mathbf{B}_\Delta$ stabilize the system. This condition can be guaranteed in flexible structure designs by restricting spring constants and damping coefficients to be positive (a natural assumption in passive structures). Applications of branch and bound to other areas of control system analysis and synthesis may be found in [1, 2, 6]. The next section briefly reviews the branch and bound algorithm and presents a concise development of a new lower bound that improves its performance.

2 Algorithm and Bounds

The branch and bound algorithm and notation are taken from [2]. The algorithm is reproduced below. The iteration index is denoted by k , the list of cubes by \mathcal{L}_k , the lower bound by L_k and the upper bound by U_k for $\Phi_{\min}(\mathbf{B}_\Delta)$ at the end of k iterations. Φ_{lb} and Φ_{ub} are lower and upper bounds, respectively, for Φ_{\min} over a given parameter rectangle, and \mathcal{Q} denotes a subrectangle in \mathbf{B}_Δ .

```

k = 0
 $\mathcal{L}_0 = \{\mathbf{B}_\Delta\};$ 
 $L_0 = \Phi_{\text{lb}}(\mathbf{B}_\Delta);$ 
 $U_0 = \Phi_{\text{ub}}(\mathbf{B}_\Delta);$ 
if  $U_k - L_k < \epsilon$ ; stop; end if
while  $U_k - L_k > \epsilon$ , {
    choose  $\mathcal{Q} \in \mathcal{L}_k$  such that  $\Phi_{\text{lb}} = L_k$ ;
    split  $\mathcal{Q}$  along its longest edge into  $\mathcal{Q}_I$  and  $\mathcal{Q}_{II}$ ;
     $\mathcal{L}_{k+1} := (\mathcal{L}_k - \{\mathcal{Q}\}) \cup \{\mathcal{Q}_I, \mathcal{Q}_{II}\};$ 
     $L_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{lb}}(\mathcal{Q});$ 
     $U_{k+1} := \min_{\mathcal{Q} \in \mathcal{L}_{k+1}} \Phi_{\text{ub}}(\mathcal{Q});$ 
    k = k + 1;
}
    
```

*Graduate student researcher, control@ucla.edu. This research was supported by NASA Grant NCC2-374.

†rtm@obsidian.seas.ucla.edu

‡pack@erg.me.berkeley.edu

2.1 Bounds

The upper and lower bounds are now developed. We only consider the case where the parameters set is \mathbf{B}_Δ because loop shifting and normalization may be used to map an arbitrary parameter rectangle to \mathbf{B}_Δ . The lower bound for Φ_{\min} in (1) is developed by first considering the constant matrix case. Let $\tilde{M} \in \mathbb{C}^{(n_y+m) \times (n_z+m)}$ and consider the problem of obtaining a lower bound for

$$\min_{\Delta \in \mathbf{B}_\Delta} \bar{\sigma}(F_L(\tilde{M}, \Delta)), \quad (2)$$

where Δ has the same structure previously defined and \tilde{M} is partitioned as $\begin{bmatrix} \tilde{M}_{zw} & \tilde{M}_{zu} \\ \tilde{M}_{yw} & \tilde{M}_{yu} \end{bmatrix}$. We assume that $F_L(\tilde{M}, \Delta)$ is well-posed for all $\Delta \in \mathbf{B}_\Delta$. Let \mathbf{u} and \mathbf{v} be the left and right singular vectors, respectively, of the maximum singular value of \tilde{M}_{zw} (we denote $\bar{\sigma}_{zw} = \bar{\sigma}(\tilde{M}_{zw})$). We may assume $\bar{\sigma}_{zw} > 0$ otherwise $\min_{\Delta} \bar{\sigma}(M, \Delta) = 0$ by choosing $\Delta = 0$. A lower bound for (2) is

$$\begin{aligned} \min_{\Delta \in \mathbf{B}_\Delta} \bar{\sigma}(F_L(\tilde{M}, \Delta)) &\geq \min_{\Delta \in \mathbf{B}_\Delta} |\mathbf{u}^* F_L(\tilde{M}, \Delta) \mathbf{v}| \\ &= \min_{\Delta \in \mathbf{B}_\Delta} [|F_L(\mathcal{M}, \Delta)|]^{-1} \\ &= \left[\max_{\Delta \in \mathbf{B}_\Delta} |F_L(\mathcal{M}, \Delta)| \right]^{-1}, \quad (3) \end{aligned}$$

where \mathcal{M} is defined as

$$\mathcal{M} = \begin{bmatrix} \frac{1}{\bar{\sigma}_{zw}} & -\frac{1}{\bar{\sigma}_{zw}} \mathbf{u}^* \tilde{M}_{zu} \\ \frac{1}{\bar{\sigma}_{zw}} \tilde{M}_{yw} \mathbf{v} & \tilde{M}_{yu} - \frac{1}{\bar{\sigma}_{zw}} \tilde{M}_{yw} \mathbf{v} \mathbf{u}^* \tilde{M}_{zu} \end{bmatrix}. \quad (4)$$

The lower bound in (3) exploits the ‘‘directionality’’ of the \tilde{M}_{zw} block, and the expression for \mathcal{M} comes from computing the LFT inverse of $\mathbf{u}^* F_L(\tilde{M}, \Delta) \mathbf{v}$. The last line in (3) is bounded from below by computing an upper bound for $\max_{\Delta \in \mathbf{B}_\Delta} \bar{\sigma}(F_L(\mathcal{M}, \Delta))$ via the scaled Main Loop Theorem (see [7, 8]). Scaling sets that commute with Δ are

$$\begin{aligned} \mathbb{D} &:= \{ \text{diag} [D_1, \dots, D_S] : D_i \in \mathbb{C}^{r_i \times r_i}, D_i = D_i^* > 0 \} \\ \mathbb{G} &:= \{ \text{diag} [G_1, \dots, G_S] : G_i \in \mathbb{C}^{r_i \times r_i}, G_i = G_i^* \}. \end{aligned}$$

Define

$$\begin{aligned} \gamma_0 &:= \inf \left\{ \gamma > 0 : \inf_{\tilde{D}, \tilde{G}} \bar{\sigma} \left[j\tilde{G}(I + \tilde{G}^2)^{-\frac{1}{2}} \right. \right. \\ &\quad \left. \left. + (I + \tilde{G}^2)^{-\frac{1}{2}} \tilde{D}_L \mathcal{M} \tilde{D}_R \right] < 1 \right\}, \quad (5) \end{aligned}$$

where

$$\begin{aligned} \tilde{D}_L &:= \begin{bmatrix} \frac{1}{\sqrt{\gamma}} & 0 \\ 0 & D \end{bmatrix} & \tilde{D}_R &:= \begin{bmatrix} \frac{1}{\sqrt{\gamma}} & 0 \\ 0 & D^{-1} \end{bmatrix}, & D &\in \mathbb{D} \\ \tilde{G} &:= \begin{bmatrix} 0 & 0 \\ 0 & G \end{bmatrix}, & G &\in \mathbb{G}, & j &= \sqrt{-1}. \end{aligned}$$

Note that (5) may be computed via the convex optimization

$$\begin{aligned} &\text{minimize } \gamma^2 \\ &\text{subject to } \begin{cases} \mathcal{M} \begin{bmatrix} 1 & 0 \\ 0 & D \end{bmatrix} \mathcal{M}^* + \begin{bmatrix} 0 & 0 \\ 0 & jG \end{bmatrix} \mathcal{M}^* \\ -\mathcal{M} \begin{bmatrix} 0 & 0 \\ 0 & jG \end{bmatrix} < \begin{bmatrix} \gamma^2 & 0 \\ 0 & D \end{bmatrix} \\ D \in \mathbb{D}, G \in \mathbb{G} \end{cases}. \quad (6) \end{aligned}$$

Lower bounds for (2) may now be stated;

$$\begin{aligned} \min_{\Delta \in \mathbf{B}_\Delta} \bar{\sigma}(F_L(\tilde{M}, \Delta)) &\geq \begin{cases} \frac{1}{\gamma_0} & \text{if (5) is feasible} \\ 0 & \text{if (5) is infeasible} \end{cases} \quad (7) \\ &\geq \begin{cases} \max\{\bar{\sigma}_{zw} - \frac{\bar{\sigma}(\tilde{M}_{zu})\bar{\sigma}(\tilde{M}_{yw})}{1 - \bar{\sigma}(\tilde{M}_{yu})}, 0\} & \text{if } \bar{\sigma}(\tilde{M}_{yu}) < 1 \\ 0 & \text{if } \bar{\sigma}(\tilde{M}_{yu}) \geq 1 \end{cases} \quad (8) \end{aligned}$$

The lower bound in (7) exploits the structure of Δ and the fact that the parameters are real-valued. This bound is often *much* tighter than (8) but clearly requires more computation. Details on these bounds and their ordering may be found in [5]. This reference also explains how the bound in (7) exploits the directionality of the \tilde{M}_{zw} block.

A lower bound for (1) is now derived from the constant matrix case. For any $\Delta \in \mathbf{B}_\Delta$ and $\omega \in \mathbb{R}$

$$\|F_L(M, \Delta)\|_\infty \geq \bar{\sigma}(F_L(M(j\omega), \Delta)).$$

Thus,

$$\min_{\Delta \in \mathbf{B}_\Delta} \|F_L(M, \Delta)\|_\infty \geq \min_{\Delta \in \mathbf{B}_\Delta} \bar{\sigma}(F_L(M(j\omega), \Delta)).$$

The uniform continuity condition between Φ_{lb} and Φ_{ub} is satisfied by choosing $\omega = \omega_0$ where $\|M_{zw}\|_\infty = \bar{\sigma}(M_{zw}(j\omega_0))$; i.e., the norm is *achieved* at ω_0 (this is not difficult to guarantee in practice because the search over ω may be limited to a bounded interval, or low-pass weighting functions on y can render the closed-loop system strictly proper). Thus a lower bound for (1) is

$$\Phi_{\text{lb},1}(\mathbf{B}_\Delta) = \begin{cases} \frac{1}{\gamma_0} & \text{if (6) is feasible} \\ 0 & \text{if (6) is infeasible} \end{cases}, \quad (9)$$

where \mathcal{M} is defined according to (4) with $\tilde{M} = M(j\omega_0)$. A computationally inexpensive lower bound for (1) is

$$\begin{aligned} \Phi_{\text{lb},2}(\mathbf{B}_\Delta) &= \\ &\begin{cases} \max\{\bar{\sigma}_{zw} - \frac{\bar{\sigma}(\tilde{M}_{zu})\bar{\sigma}(\tilde{M}_{yw})}{1 - \bar{\sigma}(\tilde{M}_{yu})}, 0\} & \text{if } \bar{\sigma}(\tilde{M}_{yu}) < 1 \\ 0 & \text{if } \bar{\sigma}(\tilde{M}_{yu}) \geq 1, \end{cases} \quad (10) \end{aligned}$$

where $\tilde{M} = M(j\omega_0)$ again (note: $\bar{\sigma}_{zw} = \|M_{zw}\|_\infty$ by the choice of ω_0). A branch and bound performance comparison between bounds (9) and (10) is made in Section 3.

Finally, the upper bound for (1), Φ_{ub} , is quite simple:

$$\Phi_{ub} := \|F_L(M, 0)\|_\infty = \|M_{zw}\|_\infty. \quad (11)$$

In other words, the norm of the system closed with the parameters at the center of \mathbf{B}_Δ provides an upper bound for Φ_{min} .

2.2 Continuity Condition

The continuity condition that guarantees convergence of the algorithm is

$$\begin{aligned} \forall \epsilon > 0 \exists \delta > 0 \forall Q \subseteq \mathbf{B}_\Delta \text{ size}(Q) \leq \delta \\ \implies \Phi_{ub}(Q) - \Phi_{lb,i}(Q) \leq \epsilon \end{aligned} \quad (12)$$

where $\text{size}(Q)$ denotes the length of the longest side of Q .

The upper bound in (11) is the same as that used in [2] where it was shown that Φ_{ub} and

$$\Phi_{lb,3}(\mathbf{B}_\Delta) =$$

$$\begin{cases} \max\{\|M_{zw}\|_\infty - \frac{\|M_{zu}\|_\infty \|M_{yw}\|_\infty}{1 - \|M_{yu}\|_\infty}, 0\} & \text{if } \|M_{yu}\|_\infty < 1 \\ 0 & \text{if } \|M_{yu}\|_\infty \geq 1 \end{cases} \quad (13)$$

satisfy (12). Thus, in order to demonstrate the continuity of Φ_{ub} and $\Phi_{lb,1}$, and of Φ_{ub} and $\Phi_{lb,2}$, it suffices to show $\Phi_{lb,2} \geq \Phi_{lb,3}$ since $\Phi_{lb,1} \geq \Phi_{lb,2}$.

Comparing (10) and (13), however, it is evident that $\Phi_{lb,2} \geq \Phi_{lb,3}$, since

$$\begin{aligned} \|M_{zu}\|_\infty &\geq \bar{\sigma}(\tilde{M}_{zu}), \quad \|M_{yw}\|_\infty \geq \bar{\sigma}(\tilde{M}_{yw}), \\ \|M_{yu}\|_\infty &\geq \bar{\sigma}(\tilde{M}_{yu}). \end{aligned}$$

3 Design Examples

The branch and bound algorithm with the lower bounds (9) and (10) are applied to two flexible structure design problems. The objective is to minimize the \mathcal{H}_∞ norm of the structural response from disturbance forces acting at different points in the structure. The requirement that all $\Delta \in \mathbf{B}_\Delta$ stabilize the system is ensured by constraining the stiffness and damping matrices to be positive definite in the structural equations. This is guaranteed by specifying the parameter intervals such that the individual spring rates and damping

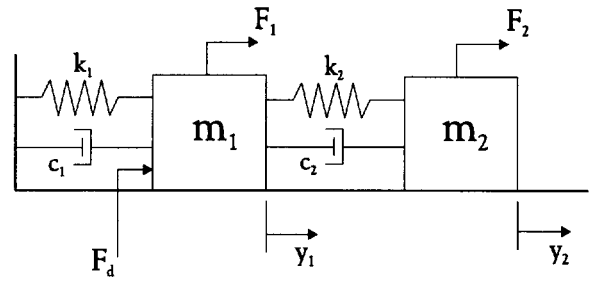


Figure 1: Vibration absorption system

coefficients are positive (although it doesn't imply the design parameters are necessarily positive since the intervals are relative to the nominal spring and damping values).

The branch and bound algorithm was terminated when $\Phi_{lb,i} \geq 0.9\Phi_{ub}$. As one would expect, the tighter the stopping criteria, the more computationally intensive the problem becomes. It is up to the designer to determine the tradeoff in tightness of the bounds versus computation time. All simulations were performed on a Pentium II 400 MHz machine using Matlab. The LMI toolbox [4] was used to compute γ_0 in (9).

3.1 Dynamic Vibration Absorber

This first example is the design of a vibration absorber for the two mass system shown in Figure 1. The objective is to attenuate the velocity of the first mass due to the disturbance force F_d at a specific frequency by designing k_2 and c_2 appropriately. The weighting function for y_1 is

$$W_{y_1} = \frac{10s}{s^2 + \frac{1}{50}s + 2}.$$

This weight places a large peak in the "closed-loop" transfer function at $\sqrt{2} \frac{\text{rad}}{\text{sec}}$. By minimizing the \mathcal{H}_∞ gain from F_d to weighted output, the algorithm will attempt to minimize the peak of the disturbance-to-velocity Bode plot across frequency, and in doing so will force the unweighted system to exhibit a notch at $\sqrt{2} \frac{\text{rad}}{\text{sec}}$.

The physical constants are chosen as $m_1 = m_2 = 1$, $k_1 = c_1 = 1$, and k_2 and c_2 are set at nominal values of 1 as well. The initial parameter ranges are set to $\delta_{k_2} \in [-0.99, 5]$ and $\delta_{c_2} \in [-0.99, 5]$. These parameter ranges imply that the algorithm searches the intervals $k_2 \in [0.01, 6]$ and $c_2 \in [0.01, 6]$. Note that for this simple example we can select the optimal parameters by inspection: the vibration absorber effect requires that the m_2 - k_2 - c_2 subsystem exhibit lightly damped oscillations at $\sqrt{2} \frac{\text{rad}}{\text{sec}}$ when y_1 is fixed; this implies that $k_2 \approx 2$ and $c_2 \approx 0$.

Using the tight lower bound $\Phi_{lb,1}$, the algorithm com-

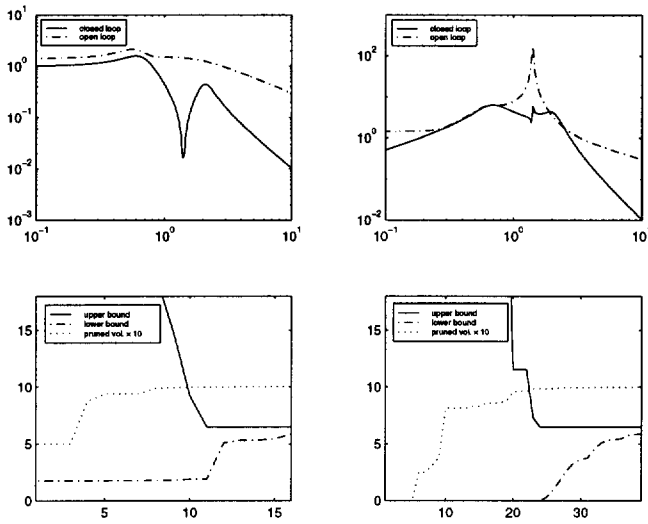


Figure 2: Singular value plots for weighted system (top right) and unweighted system (top left) and progress with $\Phi_{lb,1}$ (bottom left) and $\Phi_{lb,2}$ (bottom right).

pleted the optimization in 16 iterations with a total time of 7.4 seconds, and $\Phi_{ub} = 6.484$ and $\Phi_{lb,1} = 5.858$. For comparison, this algorithm with lower bound $\Phi_{lb,2}$ required 39 iterations and a total time of 9.0 seconds with $\Phi_{ub} = 6.483$ and $\Phi_{lb,2} = 5.889$. This is not a dramatic difference between the two cases, but as the number of parameters is increased, the gaps between the total number of iterations and total times increase significantly.

The final parameters found by the algorithm using $\Phi_{lb,1}$ are $\delta_{k_2} = 0.9663$ and $\delta_{c_2} = -0.9804$ and using $\Phi_{lb,2}$ are $\delta_{k_2} = 0.9667$ and $\delta_{c_2} = -0.9789$. These correspond to $k_{2,opt} \approx 1.97$ and $c_{2,opt} \approx 0.02$, which are essentially the analytically determined values. The Bode plots of the weighted and unweighted, open and closed-loop systems are shown Figure 2. The notching effect can be seen in the unweighted closed-loop plot. The progress on the upper and lower bounds and pruned volume versus iteration are also shown for both bounds. Note that the pruned volume is multiplied by a factor of 10 so that it can be displayed on the same graph as the bounds. The algorithm with the tight lower bound is able to prune significant amounts of volume from the start. The initial lower bound is also nonzero for this case.

3.2 Five degree of freedom system

The system for this design example is in Figure 3. The objective is to minimize the \mathcal{H}_∞ norm from the F_{d_1} and F_{d_2} disturbance forces to the (unweighted) velocities of masses m_2 and m_4 . The physical constants are $m_i = 1$, $k_i = 1$, $c_1 = c_3 = c_5 = 0$, and $c_2 = c_4 = 0.01$.

The first design uses k_2 and c_2 as the parameters, the second design uses k_2 , c_2 and c_4 as the parameters, and the third design uses k_2 , c_2 , k_4 and c_2 as the parameters. The spring constant ranges are $[0.01, 6]$ (so $\delta_{k_i} \in [-0.99, 5]$) and the damping coefficient ranges are $[0.01, 5.01]$ (so $\delta_{c_i} \in [0, 5]$).

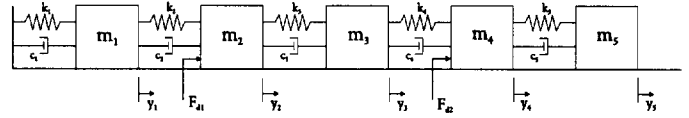


Figure 3: 5 mass system

In the two parameter case the optimization finished in 29 iterations with a final time of 12.7 seconds using the tight lower bound $\Phi_{lb,1}$. With the bound $\Phi_{lb,2}$, the optimization required 11,331 iterations and a total time of 50.9 minutes. Here we see a significant difference in algorithm performance even with two parameters. The tight lower bound was nonzero on the first iteration, however it took 10,464 iterations before “crude” lower bound was able to achieve a nonzero value. The final parameters using $\Phi_{lb,1}$ are $\delta_{k_2} = -0.7092$ and $\delta_{c_2} = 0.5496$, and $\Phi_{ub} = 10.524$ and $\Phi_{lb,1} = 9.696$. The final parameters using $\Phi_{lb,2}$ are $\delta_{k_2} = -0.7092$ and $\delta_{c_2} = 0.5469$, and $\Phi_{ub} = 10.525$ and $\Phi_{lb,2} = 9.479$.

In the three and four parameter cases, it was not possible to run the optimization to completion using the vertically challenged bound, $\Phi_{lb,2}$. Each case was run for 12 hours and then terminated. At that point, the lower bounds were still zero. The three parameter optimization with $\Phi_{lb,1}$ finished in 675 iterations and 334.8 seconds, and $\Phi_{ub} = 5.794$ and $\Phi_{lb,1} = 5.219$. The four parameter required 6661 iterations and 1.22 hours, and $\Phi_{ub} = 3.083$ and $\Phi_{lb,1} = 2.775$ —an obvious improvement over the branch and bound performance with bound $\Phi_{lb,2}$.

The optimum values for the three parameter case are $\delta_{k_2} = -0.8496$, $\delta_{c_2} = 0.5469$, and $\delta_{c_4} = 0.0781$ and for the four parameter case are $\delta_{k_2} = -0.2646$, $\delta_{k_4} = -0.9666$, $\delta_{c_2} = 0.0977$, and $\delta_{c_4} = 0.4883$. Figure 4 shows the maximum singular value plot of the initial system and the three “closed-loop” systems. As one would expect, as the number of design parameters is increased, the optimization is able to push the peak response to a lower level. The final spring rates and damping values are also markedly different in each case.

Lastly, we compare the branch and bound performance on the structure design problems versus its performance on the constant matrix problem in (2). The

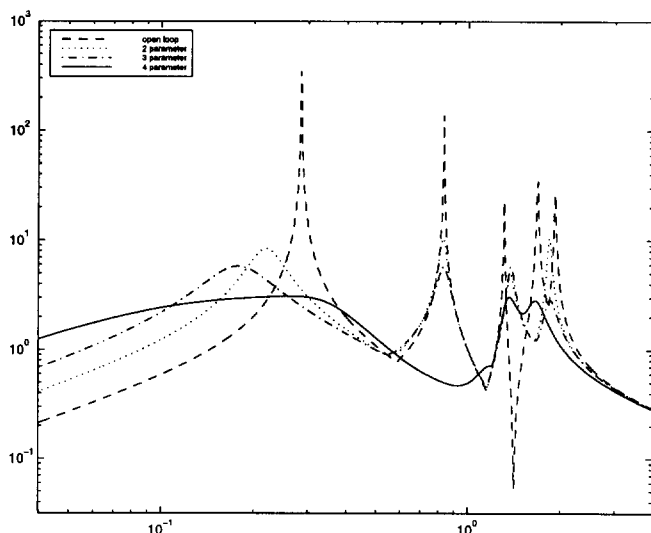


Figure 4: Singular value plots of optimal designs for 2-, 3-, and 4-parameter cases.

constant matrix case runs significantly faster than the system designs. A summary of the average run times for 50 random complex matrices (for each parameter case) is shown in the following tables.

# param.	Constant Matrix		
	2	3	4
with $\Phi_{lb,1}$	0:0:1*	0:0:4	0:0:19
with $\Phi_{lb,2}$	0:0:3	0:0:34	0:11:29

* - all times are hrs:mins:secs

#param.	5-DOF System Design			Absorber
	2	3	4	
with $\Phi_{lb,1}$	0:0:13*	0:5:35	1:13:6	0:0:7
with $\Phi_{lb,2}$	0:50:55	>12hrs	>12hrs	0:0:9

* - all times are hrs:mins:secs

4 Conclusion

The application of branch and bound to minimizing the \mathcal{H}_∞ norm of an LFT is presented. We demonstrate that sharp lower bounds are essential for reasonable total computation times. Even though tighter lower bounds may consume more computation time than simple bounds, an overall savings in the number of iterations reduces the total time. This increased efficiency is amplified as the number of optimization parameters increases. The results are illustrated on two optimal structure designs. We also observed that the computation time depends on the specific choice of design parameters and not just the number of parameters.

A comparison between the optimization of dynamical systems and constant matrices is also explored. It seems that the added complexity of the dynamic case can significantly increase computation time. One possible explanation is that the algorithm must concentrate on refining several regions of the parameter space in the system design scenario. This is exacerbated by the fact that the \mathcal{H}_∞ norm is achieved at several frequencies in the optimal design (this is evident from the maximum singular value plots in Figure 4).

References

- [1] Balakrishnan, V., Boyd, S., and Balemi, S., "Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems," *International Journal of Robust and Nonlinear Control*, Vol. 1, No. 4, 1991, pp. 295-317.
- [2] V. Balakrishnan and S. Boyd, "Global optimization in control system analysis and design," in *Advances in Control Systems*, C.T. Leondes, ed., Academic Press, New York, 1992.
- [3] Balas, G. J., Doyle, J. C., Glover, K., Packard, A., and Smith, R., *μ -Analysis and Synthesis Toolbox*, The MathWorks, Inc., 1997.
- [4] Gahinet, P., Nemirovski, A., Laub, A., and Chilali, M., *LMI Control Toolbox*, The Math Works, Natick, MA, 1995.
- [5] M'Closkey, R., Packard, A., and Sipila, J. D., "Branch and bound computation of the minimum norm of a linear fractional transformation over a structured set," submitted, August 1998.
- [6] Newlin, M.P., and Young, P.M., "Mixed mu problems and branch and bound techniques," *International Journal of Robust and Nonlinear Control*, Vol. 7, No. 2, 1997, pp. 145-64.
- [7] Packard, A., and Doyle, J. C., "The complex structured singular value," *Automatica*, Vol. 29, 1993, pp. 71-109.
- [8] K. Zhou with J. Doyle, *Essentials of Robust Control*, Prentice Hall, New Jersey, 1998.