A Path-following Method for Solving BMI Problems in Control

Arash Hassibi[1] Jonathan How Stephen Boyd

Information Systems Laboratory
Stanford University
Stanford, CA 94305-9510, USA

Abstract—In this paper we present a path-following (homotopy) method for (locally) solving bilinear matrix inequality (BMI) problems in control. The method is to linearize the BMI using a first order perturbation approximation, and then iteratively compute a perturbation that "slightly" improves the controller performance by solving a semidefinite program (SDP). This process is repeated until the desired performance is achieved, or the performance cannot be improved any further. While this is an approximate method for solving BMIs, we present several examples that illustrate the effectiveness of the approach.

Keywords: Bilinear matrix inequality (BMI), linear matrix inequality (LMI), semidefinite programming (SDP), robust control, low-authority control.

1 Introduction

Promising new methods for the analysis and design of robust controllers for linear and nonlinear uncertain systems have emerged over the last several years. The basic idea is to formulate the analysis or synthesis problem in terms of convex or bi-convex matrix optimization problems which are then solved numerically. Most of this research has concentrated on the *semidefinite programming problem* (SDP), *i.e.*, the problem of minimizing a linear cost function over *linear matrix inequalities* (LMIs). SDPs are convex optimization problems that can be solved with great practical and theoretical efficiency using interior-point algorithms [1, 2, 3, 4, 5].

Other control problems, including synthesis with structured uncertainty, fixed-order controller design, output feedback stabilization, simultaneous stabilization, decentralized controller synthesis, etc., lead to *bilinear matrix inequalities* (BMIs). See, for example [6, 7, 8]. BMI problems are not convex and can have multiple local solutions. The computational complexity for solving BMI problems is much higher than LMI problems so researchers have been looking at a variety of iterative schemes to solve them locally. One well-known scheme is to alternate between analysis and synthesis via LMIs that often results in acceptable local solutions. For global (branch and bound) methods for solving BMI problems refer to [6, 9].

In this paper we present a path-following (homotopy) method for (locally) solving BMI problems in control. The method is very easy to implement: the BMI is linearized using a first order perturbation approximation, and then a perturbation is computed that "slightly" improves the controller performance by solving an SDP. This process is repeated until the desired performance is achieved, or the performance cannot be improved any further.

2 Linearization method for solving BMIs in "low-authority" control

The idea of solving BMIs by linearization and SDP has been used in the context of low-authority controller (LAC) design [10, 11]. The assumption in LAC is that the actuators have "limited authority" and hence the performance of the closed-loop and open-loop systems are "close". Therefore, using first order perturbation formulas, it is possible to predict the performance of the closed-loop system accurately. As a result, many control problems that are normally intractable and require the solution to BMIs can be formulated as LMIs which can then be solved very efficiently.

In order to illustrate this linearization method, consider the problem of linear output-feedback design with limits on the feedback gains. Specifically, consider the linear time-invariant dynamical system with input and output

$$\dot{x} = Ax + Bu, \quad y = Cx$$

where the open-loop system $\dot{x} = Ax$ has a damping or decay rate of at least α. The goal is to design the feedback gain matrix $\delta K \in \mathbf{R}^{m\times n}$ such that the control law $u = \delta K y$ gives an additional damping of $\delta\alpha$ in the closed-loop system, while the controller gains satisfy the interval constraints

$$|\delta K_{ij}| \le l_{ij},$$

for $i = 1, \dots, m$, and $j = 1, \dots, n$. This problem is known to be NP-hard [12].

By simple Lyapunov theory (see, e.g., [13]), this problem is equivalent to the existence of $P \in \mathbf{SR}^{n\times n}$ such that

$$\begin{array}{l} P \succ 0, \quad |\delta K_{ij}| \le l_{ij}, \\ (A + B\delta KC)^T P + P(A + B\delta KC) \preceq -2(\alpha + \delta\alpha)P, \end{array} \tag{1}$$

which is a BMI in the variables P and δK.

The linearization method for solving the BMI (1) can be explained as follows. Since the open-loop system has a decay rate of at least α, it is possible to compute $P_0 \succ 0$ such that

$$A^T P_0 + P_0 A \preceq -2\alpha P_0. \tag{2}$$

Now write $\delta P = P - P_0$ so that (1) becomes

$$\begin{array}{l} P_0 + \delta P \succ 0, \quad |\delta K_{ij}| \le l_{ij}, \\ (A + B\delta KC)^T (P_0 + \delta P) + (P_0 + \delta P)(A + B\delta KC) \\ \qquad \preceq -2(\alpha + \delta\alpha)(P_0 + \delta P), \end{array} \tag{3}$$

Under the low-authority assumption it is reasonable to assume that δP, $\delta\alpha$, and δK are "small", and therefore their product is to first order negligible. Hence by neglecting the second order terms $\delta P B \delta K C$, $C^T \delta K^T B^T \delta P$, and $\delta\alpha\delta P$ in (3) we get

$$\begin{array}{l} P_0 + \delta P \succ 0, \quad |\delta K_{ij}| \le l_{ij}, \\ A^T(P_0 + \delta P) + (P_0 + \delta P)A + P_0 B\delta KC + \\ \qquad C^T \delta K^T B^T P_0 \prec -2\alpha(P_0 + \delta P) - 2\delta\alpha P_0. \end{array} \tag{4}$$

[1] Contact author. E-mail: arash@isl.stanford.edu. Research supported by Air Force (under F49620-97-1-0459), AFOSR (under F49620-95-1-0318), and NSF (under ECS-9222391 and EEC-9420565).

Clearly, (4) is an LMI in the variables δP and δK which can be solved efficiently for the desired feedback gain matrix δK. Of course, once (4) has been solved one should go back and check if the low-authority assumption was correct, *i.e.*, the linearization error was negligible (otherwise, more iterations of LAC design are required). References [10, 11] provide several examples that illustrate this design procedure.

Note that this linearization method is quite powerful and can also be applied to many other problems such as multi-objective controller design, decentralized control, and simultaneous actuator/sensor placement and controller design.

3 Path-following method for solving BMIs in control

The linearization method for LAC design in the previous section suggests a path-following (homotopy) method for (locally) solving BMIs in control. Roughly speaking, the approach is to achieve the overall design objective by iteratively solving a sequence of linearized problems, which at each step results in a controller that is incrementally better than the previous one.

In other words, starting from the initial (open-loop) system, the idea is to design better and better controllers by slowly improving the design objective. (For example, given a reduced-order, decentralized, or fixed architecture controller we could iteratively design for lower values of induced $\mathcal{L}_2$ norm). Since the design objectives in consecutive problems are "close", at each step, we can linearize the BMI to accurately design a controller that is slightly better than the previous one by solving an SDP. Hence, the BMI is converted to a series of LMIs along a "path" parameterized by the closed-loop performance.

This path-following method can be used to heuristically solve many BMI problems in control. However, there are no convergence guarantees to an acceptable solution. As with all local methods for solving BMIs, the choice of initial value is important for convergence to an acceptable solution, which is a potential weakness of this method. For example, it is not clear which P_0 should be used in (4) among all P_0's that satisfy (2). As long as P_0 is "close enough" to the optimal P however, we conjecture that it does not make much difference which P_0 is chosen because P_0 can be adjusted iteratively using the free variable δP. (Our experience indicates that the P_0 with smallest condition number, or the one that minimizes $\log\det P_0^{-1}$, seem to work well in practice.) Therefore, this method works best for "medium-authority controller" (MAC) designs in which the required closed-loop system performance is not drastically better than the open-loop system performance.

In the next section we present examples from control for solving BMIs using this method. In each case we briefly explain the iterative method for solving the corresponding BMI and the choice of initial value. These examples show the method is very effective in solving such problems.

4 Examples

4.1 Sparse linear constant output-feedback design

Consider the BMI optimization problem

$$\begin{array}{ll} \text{minimize} & \sum_{ij} |K_{ij}| \\ \text{subject to} & P \succ 0, \\ & (A+BKC)^T P + P(A+BKC) \prec -2\alpha P \end{array} \qquad (5)$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, and $C \in \mathbf{R}^{p\times n}$ are given matrices. This corresponds to designing a *sparse* linear constant output feedback control $u = Ky$ for the system $\dot{x} = Ax + Bu$, $y = Cx$ which results in a decay rate of at least α in the closed-loop system. Minimizing the ℓ_1 norm of the feedback gains as in (5) is a good heuristic for obtaining sparse feedback gain matrices (see [10, 11]). Finding sparse feedback gain matrices is a way of solving the actuator/sensor placement or controller topology design problems.

To solve this BMI using a path-following method we proceed as follows.

1. Let $K := 0$.
2. Compute the Lyapunov matrix P_0 with minimum condition number that proves the level of decay rate α_0 in the system $\dot{x} = Ax$ (α_0 is the smallest negative real part of the eigenvalues of A). This is done by solving the SDP

$$\begin{array}{ll} \text{minimize} & \kappa \\ \text{subject to} & I \prec P_0 \prec \kappa I, \\ & A^T P_0 + P_0 A \prec -2(\alpha_0 - \epsilon) P_0, \end{array} \qquad (6)$$

where ϵ is a small positive number ($0 < \epsilon \ll 1$).

3. Solve the following SDP which is the linearized version of (5) around P_0 and K:

$$\begin{array}{ll} \text{minimize} & \sum_{ij} |K_{ij} + \delta K_{ij}| \\ \text{subject to} & P_0 + \delta P \succ 0, \quad \|\delta P\| < 0.2\|P_0\| \\ & A^T(P_0+\delta P) + (P_0+\delta P)A + \\ & P_0 B \delta K C + C^T \delta K^T B^T P_0 \prec \\ & \qquad - 2\alpha_0(P_0+\delta P) - 2\delta\alpha P_0 \end{array} \qquad (7)$$

where $\delta\alpha$ is chosen to be "small". Note that the constraint $\|\delta P\| < 0.2\|P_0\|$ is added so that the perturbation is small and the linear approximation should be valid.

4. Let $K := K + \delta K$, $A := A + B\delta KC$ and go to step 2.

The iteration in the above algorithm stops whenever α_0 exceeds the desired α, or if α_0 cannot be improved any further (*i.e.*, (7) is infeasible for any $\delta\alpha > 0$).

As an example, suppose that

$$A = \begin{bmatrix} -2.45 & -0.90 & 1.53 & -1.26 & 1.76 \\ -0.12 & -0.44 & -0.01 & 0.69 & 0.90 \\ 2.07 & -1.20 & -1.14 & 2.04 & -0.76 \\ -0.59 & 0.07 & 2.91 & -4.63 & -1.15 \\ -0.74 & -0.23 & -1.19 & -0.06 & -2.52 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.81 & -0.79 & 0.00 & 0.00 & -0.95 \\ -0.34 & -0.50 & 0.06 & 0.22 & 0.92 \\ -1.32 & 1.55 & -1.22 & -0.77 & -1.14 \\ -2.11 & 0.32 & 0.00 & -0.83 & 0.59 \\ 0.31 & -0.19 & -1.09 & 0.00 & 0.00 \end{bmatrix},$$

$$C = \begin{bmatrix} 0.00 & 0.00 & 0.16 & 0.00 & -1.78 \\ 1.23 & -0.38 & 0.75 & -0.38 & 0.00 \\ 0.46 & 0.00 & -0.05 & 0.00 & 0.00 \\ 0.00 & -0.12 & 0.23 & -0.12 & 1.14 \end{bmatrix}.$$

The system defined by (A, B, C) is unstable with a decay rate of $\alpha_0 = -0.2832$ (growth rate of 0.2832). The goal is to design a sparse K so that the decay rate of the closed-loop system $\dot{x} = (A + BKC)x$ is not less than 0.35. After 6 iterations of the path-following method with $\delta\alpha = 0.1$ and $\epsilon = 0.01$ we get a closed-loop decay rate of $\alpha_0 = 0.3543$ with

$$K = \begin{bmatrix} 0.2461 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0059 & 0.3265 & 0.0000 & 0.0000 \end{bmatrix}.$$

Clearly, the resulting K is sparse. It only has two nonzero columns (1 and 2), two nonzero rows (1 and 5), and three nonzero elements (11, 51, and 52). Hence only the first and second sensors, and the first and fifth actuators are needed. Also, the controller has simple topology since we only need to connect sensor 1 to actuator 1, and sensors 1 and 2 to actuator 5.

Thus, the optimization has succeeded in simultaneously performing the sensor/actuator placement problem and the feedback control design. The ℓ_1 minimization heuristic has done a great job noting that the number of sparsity patterns of K is $2^{20} \approx 10^6$ and an exhaustive search method would be very time-consuming if not impractical.

4.2 Simultaneous state-feedback stabilization with limits on feedback gains

Here we consider the problem of stabilizing three different linear systems using a common linear constant state-feedback law with limits on the feedback gains. Specifically, suppose that

$$\dot{x} = A_k x + B_k u, \quad u = Kx, \quad k = 1, 2, 3.$$

The goal is to compute K satisfying $|K_{ij}| \le K_{ij,\max}$ such that all three closed-loop systems

$$\dot{x} = (A_k + B_k K)x, \quad k = 1, 2, 3$$

are stable. This problem is known to be NP-hard [14].

A stabilizing feedback gain K exists if and only if the optimum of the following BMI problem is positive:

$$\begin{array}{ll} \text{maximize} & \min_k \alpha_k \\ \text{subject to} & |K_{ij}| \le K_{ij,\max}, \\ & (A_k + B_k K)^T P_k + P_k(A_k + B_k K) \prec -2\alpha_k P_k, \\ & P_k \succ 0, \quad k = 1, 2, 3. \end{array} \tag{8}$$

The path-following method for solving this BMI problem locally can be briefly explained as follows. Similar to the method of the previous example, we first compute the minimum condition number Lyapunov matrices P_k, $k = 1, 2, 3$ that prove the level of decay rate α_k for each of the three different systems (as in (6)). Next we solve the linearized version of (8) around K (initially $K = 0$), α_k, and the computed P_k's. A_k and K are updated as $K := K + \delta K$, $A_k := A_k + B_k \delta K$, and the procedure is repeated.

As an example suppose that

$$A_1 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & -0.5 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.5 & -7 & 0 \\ 7 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} -0.5 & -3 & 0 \\ 3 & -0.5 & 0 \\ 0 & 0 & 2 \end{bmatrix},$$

and

$$B_1 = B_2 = B_3 = \begin{bmatrix} 0.2477 & -0.1645 \\ 0.4070 & 0.8115 \\ 0.6481 & 0.4083 \end{bmatrix}.$$

Note that all three systems are unstable. With $K_{ij,\max} = 50$, after 15 iterations, the path-following method gives

$$K = \begin{bmatrix} -50.0000 & 23.6909 & 33.6566 \\ -3.8940 & -50.0000 & -48.8410 \end{bmatrix}, \quad \alpha = 1.05.$$

Since $\alpha > 0$ the three systems are simultaneously stabilizable and furthermore, stabilized! Note that the maximum gain condition of $K_{ij,\max} = 50$ is active.

4.3 $\mathcal{H}_2/\mathcal{H}_\infty$ controller design

Consider the system

$$\dot{x} = Ax + Bu + B_1 w, \quad z_1 = C_1 x + D_1 u, \quad z_2 = C_2 x + D_2 u.$$

The goal is to find a feedback gain matrix K such that for $u = Kx$ the $\mathcal{H}_2$ norm from w to z_2 is minimized while the $\mathcal{H}_\infty$ norm from w to z_1 is less than some prescribed level γ. This can be done by solving the BMI optimization problem (cf. [15])

$$\begin{array}{l} \text{minimize} \ \ \eta^2 \\ \text{subject to} \\ \begin{bmatrix} \begin{pmatrix} (A+BK)^T P_1 + P_1(A+BK) + \\ (C_1^T + D_1 K)^T (C_1^T + D_1 K) \end{pmatrix} & P_1 B_1 \\ B_1^T P_1 & -\gamma^2 I \end{bmatrix} \prec 0, \\ \begin{bmatrix} (A+BK)^T P_2 + P_2(A+BK) & P_2 B_2 \\ B_2^T P_2 & -I \end{bmatrix} \prec 0, \\ \begin{bmatrix} P_2 & C_2^T \\ C_2 & Z \end{bmatrix} \succ 0, \quad \mathbf{Tr}(Z) < \eta^2, \quad P_1 \succ 0, \quad P_2 \succ 0. \end{array} \tag{9}$$

A path-following method for solving this BMI is as follows.

1. Compute an initial K say by the method of [15] which can be done using SDP. This method assumes a *common* Lyapunov matrix for the $\mathcal{H}_2$ and $\mathcal{H}_\infty$ problems and is therefore suboptimal. Suppose that P_1 is the Lyapunov matrix obtained using this method that proves a level of γ in the $\mathcal{H}_\infty$ norm.
2. With $u = Kx$, compute the $\mathcal{H}_2$ norm η of the closed-loop system and corresponding Lyapunov matrix P_2.
3. Solve the linearized BMI (9) around K, P_1, η^2, and P_2 using SDP to get the perturbations δK and δP_1.
4. Let $K := K + \delta K$, $A := A + B\delta K$, $C := C + D\delta K$.
5. Solve the SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} A^T P + PA + C_1^T C_1 & PB_1 \\ B_1^T P & -\gamma^2 I \end{bmatrix} \prec 0, \\ & -tI \prec P - (P_1 + \delta P_1) \prec tI, \quad P_1 \succ 0. \end{array}$$

   This gives the Lyapunov matrix P which proves a level of γ in the $\mathcal{H}_\infty$ norm for the closed-loop system, and is closest to the first-order adjusted P_1 (in spectral norm). Let $P_1 := P$ and go to step 2.

The iteration in the above algorithm stops whenever η^2 cannot be improved any further (*i.e.*, $\delta\eta^2 = 0$ at step 3).

As an example suppose that

$$A = \begin{bmatrix} -1.40 & -0.49 & -1.93 \\ -1.73 & -1.69 & -1.25 \\ 0.99 & 2.08 & -2.49 \end{bmatrix}, \quad B = \begin{bmatrix} 0.25 \\ 0.41 \\ 0.65 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} -0.16 & -1.29 \\ 0.81 & 0.96 \\ 0.41 & 0.65 \end{bmatrix}, \quad C_1 = [-0.41 \ \ 0.44 \ \ 0.68],$$

$$C_2 = [-1.77 \ \ 0.50 \ \ -0.40], \quad D_1 = D_2 = 1,$$

and $\gamma = 2$. The method of [15] gives

$$K_{\text{kharg}} = [1.3485 \ \ -0.2865 \ \ 0.4801]$$

resulting in $\eta = 0.8389$. A couple of iterations of the path-following method reduces the $\mathcal{H}_2$ norm to $\eta = 0.3286$ with

$$K = [0.9434 \ \ -0.6514 \ \ -0.9636].$$

4.4 Joint HAC/LAC design

Consider the truss structure shown in Figure 1. The structure consists of 39 bars with stiffness and damping connecting 17 masses at the nodes. The dynamics of the structure can be written as $\dot{z} = Az$ where $A \in \mathbf{R}^{64\times 64}$, and the state variable z consists of (a linear combination) of the horizontal and vertical displacements, and rates of displacements of each mass, u_i, v_i, $\dot{u}_i$, and $\dot{v}_i$ respectively for $i = 1, \ldots, 17$.

This problem investigates a typical LAC application, which is to add modest damping to a structure to compensate for spillover from a higher-authority controller (HAC) that has been designed using a reduced model of the structure. The design methodology in this example follows the classic two-step process [16, 17]. We first design a HAC, which is, for example, a Linear-Quadratic Gaussian (LQG) controller or a controller that achieves some eigenvalue-placement specification (note that the specific design process for the initial HAC is not important for this paper). A key point is that these higher authority designs are typically based on significantly reduced order models of the system to avoid designing a very high order controller. As a result, we would expect a considerable amount of *spillover* of the control authority to the higher frequency modes of the structure. The destabilizing effects of the spillover are addressed during the second step of the design by adding sufficient damping along bars. In this example it is required that the additional damping be such that the closed-loop eigenvalues fall within the shaded region of Figure 2 (corresponding to a minimum damping of 0.01 and minimum damping ratio of 0.02).

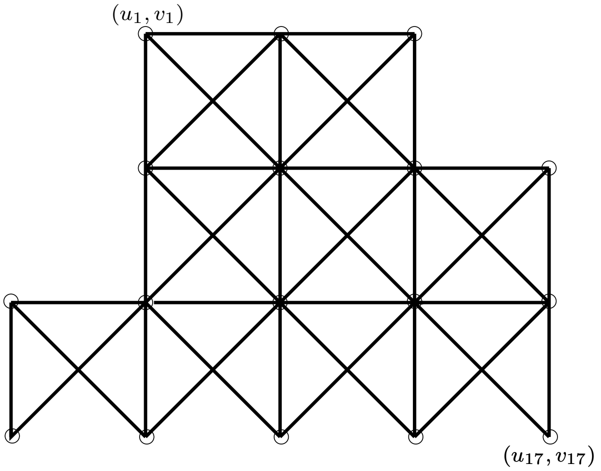

Figure 1: Truss structure consists of 39 bars (stiffness and damping) and 17 nodes (masses).

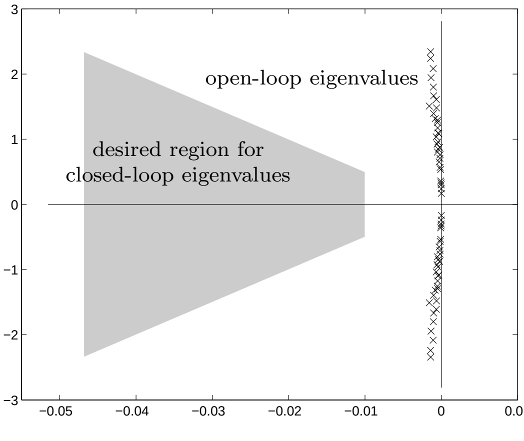

Figure 2: Open-loop eigenvalues of structure and the desired region for closed-loop eigenvalues.

The full-order HAC controller is designed based on a reduced-order truss model, using the 5 lowest frequency modes. These are the most lightly damped modes and have a good frequency seperation from the remaining 27 modes. The eigenvalues of the feedback interconnection of the HAC (which is of order 10) and the truss are shown in the top of Figure 3 (only eigenvalues near the origin are shown). Note that, as a result of the spillover, the system is actually unstable and the eigenvalue-placement specification is obviously violated.

At the second step of the design, to satisfy the eigenvalue specifications, damping (limited in this case for illustrative purposes to a maximum size of 0.08) is added along the bars. However, with the limited amount of damping allowed in this problem, the eigenvalue-placement specifications still cannot be achieved when all dampings are set to the maximum of 0.08 (Figure 3). Therefore, besides adding damping along bars, we need to adjust the HAC to hopefully get a feasible solution. The problem of jointly designing the dampers and adjusting the HAC is a BMI which we will attempt to solve using the path-following method of this paper.

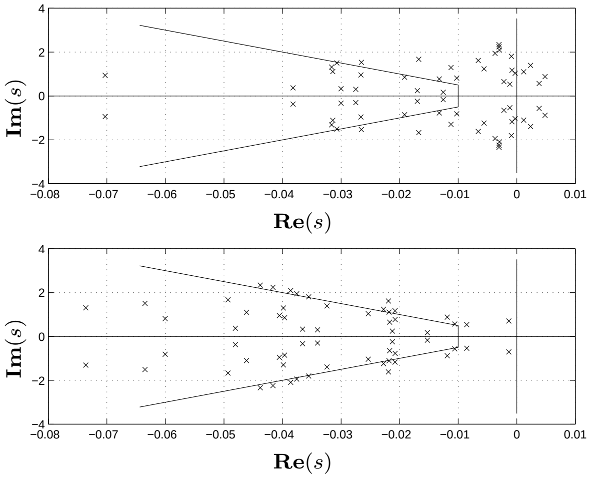

Figure 3: Eigenvalues of the feedback interconnection of the truss and initial HAC with no dampers (*top*), and with all dampers set to the maximum value of 0.08 (*bottom*). Clearly, eigenvalue specifications are not satisfied even when the dampings are maximum.

In the framework considered in this paper, for the second step of the design, the open-loop system is actually the interconnection of the HAC and the truss. The variables in the design are the amount of damping along the bars as well as perturbations to the elements of the HAC system matrices. The perturbations are limited to 5% of their original values for first order perturbation formulas to be approximately valid (the HAC is put in modal form). The problem specification is to minimize the sum of the damper values subject to the eigenvalue specifications. Using a path-following method for solving this BMI and first order perturbation formulas for the eigenvalues of a matrix [10], at each iteration we need to solve a linear program (LP) with 419 variables and 483 linear inequality constraints. This can be readily done using widely available software for solving LPs[1]

It turns out that after a single iteration of the path-following method the eigenvalue-placement specification is

[1] For example the LP solver `PCx` can be downloaded from WWW at URL `http://www-c.mcs.anl.gov/home/otc/Library/PCx/`

achieved (in other words the low-authority assumption is valid). The eigenvalue locations for the closed-loop system after adding the dampers and adjusting the HAC are shown in Figure 4. The figure clearly shows that this combined HAC/LAC design has sufficiently damped the (unstable) modes of the system. (Note that a couple of eigenvalues slightly violate the damping ratio constraint but we can simply perform another iteration to fix this problem.) Figure 5 shows the location of the nonzero dampings. The total amount of damping added to the structure in these 19 struts is 1.28 (this is less than the maximum amount of 39 × 0.08 = 3.12 tried before).

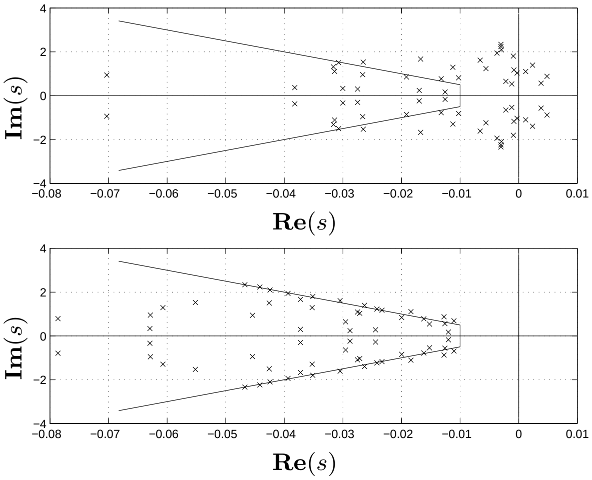

Figure 4: Eigenvalues of the feedback interconnection of the truss and HAC before adding dampers (*top*), and after adding dampers and adjusting the HAC (*bottom*).

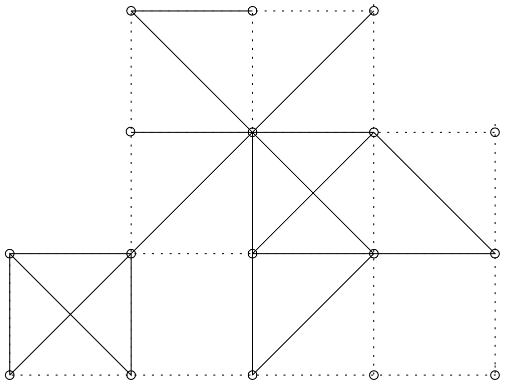

Figure 5: Location of dampers for damper design for the feedback interconnection of the plant and the HAC. A solid line between two nodes corresponds to a nonzero damper between those two nodes.

This simple problem shows that there are often key advantages to simultaneously designing the HAC and LAC components of the control architecture. More importantly, however, this example also shows that this entire control design problem can be posed as an LP, which can be solved very efficiently and very quickly on a simple computer.

5 Conclusions

In this paper we presented a path-following method for (locally) solving BMIs. The method is very easy to implement and is based on linearizing the BMIs and solving a sequence of SDPs. In general, as with all local methods for solving BMIs, the choice of initial value is important for convergence to an acceptable solution. As long as the initial value is "close enough" to the optimum value we expect the method to work well. However, the examples demonstrate that quite large performance improvements are possible using this method. It was also shown that by minimizing the ℓ_1 norm of feedback gains we can arrive at sparse designs, and therefore in effect, we can solve sensor/actuator placement and controller structure design problems.

References

[1] L. Vandenberghe and S. Boyd. SP*: Software for Semidefinite Programming. User's Guide, Beta Version.* Stanford University, October 1994. Available at `http://www-isl.stanford.edu/people/boyd`.

[2] S.-P. Wu and S. Boyd. SDPSOL: *A Parser/Solver for Semidefinite Programming and Determinant Maximization Problems with Matrix Structure. User's Guide, Version Beta.* Stanford University, June 1996.

[3] P. Gahinet and A. Nemirovskii. *LMI Lab: A Package for Manipulating and Solving LMIs.* INRIA, 1993.

[4] F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, and M. L. Overton. SDPPACK *User's Guide, Version 0.8 Beta.* NYU, June 1997.

[5] B. Borchers. CSDP*, a C library for semidefinite programming.* New Mexico Tech, March 1997.

[6] K. C. Goh, M. G. Safonov, and G. P. Papavassilopoulos. A global optimization approach for the BMI problem. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, December 1994.

[7] K.-C. Goh. *Robust Control Synthesis Via Bilinear Matrix Inequalities.* PhD thesis, University of Southern California, May 1995.

[8] M. Mesbahi, G. P. Papavassilopoulos, and M. G. Safonov. Matrix cones, complementarity problems and the bilinear matrix inequality. In *Proc. IEEE Conf. on Decision and Control*, pages 3102–3107, 1995.

[9] E. Beran, L. Vandenberghe, and S. Boyd. A global BMI algorithm based on the generalized Benders decomposition. In *Proceedings of the European Control Conference*, July 1997.

[10] A. Hassibi, J. How, and S. Boyd. Low-authority controller design via convex optimization. In *Proc. IEEE Conf. on Decision and Control*, December 1998.

[11] A. Hassibi, J. How, and S. Boyd. Low-authority controller design via convex optimization. *AIAA J. Guidance, Control, and Dynamics*, 1998. Submitted.

[12] V. D. Blondel and J. N. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM J. on Control and Optimization*, 35:2118–27, 1997.

[13] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics.* SIAM, Philadelphia, PA, June 1994.

[14] A. Nemirovskii. Several NP-Hard problems arising in robust stability analysis. *Mathematics of Control, Signals, and Systems*, 6(1):99–105, 1994.

[15] P. P. Khargonekar and M. A. Rotea. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control: A convex optimization approach. *IEEE Trans. Aut. Control*, 36(7):824–37, July 1991.

[16] J. N. Aubrun. Theory of control of structures by low-authority controllers. *J. Guidance and Control*, 3(5):444–451, September 1980.

[17] J. N. Aubrun, N. K. Gupta, and M. G. Lyons. Large space structures control: An integrated approach. In *Proc. AIAA Guidance and Control Conference*, Boulder, Colorado, August 1979.