JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 22, No. 6, November–December 1999

Low-Authority Controller Design by Means of Convex Optimization

Arash Hassibi,* Jonathan P. How,† and Stephen P. Boyd‡
Stanford University, Stanford, California 94305-9510

The premise in low-authority control is that the actuators have limited authority and hence cannot significantly shift the eigenvalues of the system. As a result, the closed-loop eigenvalues can be well approximated analytically by perturbation theory. These analytical approximations may suffice to predict the behavior of the closed-loop system in practice. We show that such approximations can be used to cast low-authority controller design problems for different objectives as convex optimization problems that can be solved efficiently in practice by using recently developed interior-point methods. Also, we show that, by optimizing the l_1 norm of the feedback gains, we can arrive at sparse designs, i.e., designs in which only a small number of the control gains are nonzero. Thus, in effect, we can also solve actuator/sensor placement or controller architecture design problems. Examples are also given that demonstrate the effectiveness of the design method.

I. Introduction

THE premise in low-authority control (LAC) is that the actuators have limited authority, and hence cannot significantly shift the eigenvalues of the system.[1,2] As a result, the closed-loop eigenvalues can be well approximated analytically by perturbation theory. These analytical approximations may suffice to predict the behavior of the closed-loop system in practical cases and will provide at least a very strong rationale for the first step in the design iteration loop.

An important use of LAC is in lightly damped large structures with an infinite number of elastic modes, for which LAC is used to provide a small amount of damping in a wide range of modes for maximum robustness. A high-authority controller (HAC) is then used around the LAC to achieve high damping or mode-shape adjustment in a selected number of modes to meet performance requirements.

In this paper we introduce a new method for low-authority controller design, based on convex programming. We formulate the LAC design problem as a nonlinear convex optimization problem, which can then be solved efficiently by interior-point methods. The advantage of formulating the problem as convex is that large-order problems can be solved (globally) in practice. Another advantage of this formulation is that it can handle a wide variety of specifications and objectives beyond standard eigenvalue placement. Typical design objectives for the LAC design include increased damping or decay rate for the system response, and typical constraints include limitations on the controller gains and actuator power. We show that, by optimizing the l_1 norm of the gains, we can arrive at sparse designs, i.e., designs in which only a small number of the control gains are nonzero. Thus, in effect, we can also solve actuator/sensor placement or controller architecture design problems. Moreover, it is possible to address the robustness of the LAC, i.e., closed-loop performance subject to uncertainties or variations in the plant model. Therefore, by combining all these, for example, we can solve the problem of robust actuator/sensor placement and LAC design in one step.

Although LAC design has been traditionally used for eigenvalue placement, when powerful Lyapunov methods are used it is possible to extend LAC design to specifications beyond eigenvalue placement. These include bounds on output energy, quadratic costs on the state and control input, induced $\mathcal{L}_2$ gain, etc.

The paper is organized as follows. Section II poses the problem statement, followed by Sec. III that presents typical applications of LAC. Section IV is a brief overview of convex programming, and in particular, linear, second-order cone, and semidefinite programming. Section V discusses the first-order perturbation formulas for the matrix eigenvalues and how the design problem can be posed within convex optimization framework. Section VI discusses the sparsity of the solution, which is important for the control architecture studies. Section VII addresses robust LAC design, i.e., a LAC design that guarantees performance subject to uncertainties or variations in the plant model. Section VIII introduces an extension to LAC design based on Lyapunov methods, and it is shown how additional performance objectives (other than eigenvalue placement) can be included in the formulation. Finally, Sec. IX demonstrates the application of the methods on a few example problems.

II. Problem Statement

We consider the linear time-invariant system

$$\dot{z} = A(\boldsymbol{x})z, \qquad z(0) = z_0 \tag{1}$$

where $z(t) \in R^n$ is the state, $\boldsymbol{x} \in R^q$ is a (design) parameter, and $A(\boldsymbol{x}) \in R^{n\times n}$ is differentiable at $\boldsymbol{x} = 0$. The goal is to find $\boldsymbol{x}$ so that the system has sufficient damping or, more generally, the eigenvalues of the system are in some desired region of the complex plane. However, it is assumed that there is "limited authority" in designing $\boldsymbol{x}$ so that the eigenvalues of system (1) are only slightly different from the eigenvalues of the unperturbed system

$$\dot{z} = A(0)z, \qquad z(0) = z_0 \tag{2}$$

i.e., system (1) with $\boldsymbol{x} = 0$. Therefore, first-order perturbation methods can be used to predict the eigenvalue locations of system (1) from the eigenvalue locations of system (2). We refer to (1) and (2) as the closed-loop and the open-loop systems, respectively.

In many applications, it is desirable to achieve the required eigenvalue locations (or damping) when $\boldsymbol{x}$ has the minimum number of nonzero elements. In such cases, each nonzero $\boldsymbol{x}_i$ may correspond to a sensor, an actuator, a dissipating mechanism, or a structural component; therefore, reducing the number of nonzero $\boldsymbol{x}_i$s simplifies the implementation. Hence we also address the problem of minimizing the number of nonzero elements of $\boldsymbol{x}$ such that the eigenvalues of system (1) are in some desired region of the complex plane.

In addition, we consider robust LAC design, i.e., a LAC design with guaranteed closed-loop system performance subject to uncertainties or variations in the system, as well as LAC design for performance measures beyond eigenvalue placement.

Received 13 July 1998; revision received 8 April 1999; accepted for publication 21 April 1999.

*Ph.D. Candidate, Department of Electrical Engineering, Information Systems Laboratory; arash@isl.stanford.edu.

†Assistant Professor, Department of Aeronautics and Astronautics, Information Systems Laboratory; howjo@sun-valley.stanford.edu. Senior Member AIAA.

‡Professor, Department of Electrical Engineering, Information Systems Laboratory; boyd@isl.stanford.edu.

III. Applications of Low-Authority Control

A key control design methodology for flexible systems with many elastic modes follows the two-level architecture presented in Refs. 1–3. This architecture consists of a wideband LAC and a narrowband HAC. Within this framework, the HAC is designed based on a (low-order) finite-dimensional model of the structure and provides high damping or mode-shape adjustment in a selected number of modes to meet performance requirements. However, because of spillover, the HAC can destabilize modes not included in the design model, which are usually at high frequency and poorly known. LAC, on the other hand, introduces low damping in a wide range of modes for maximum robustness. LAC is therefore necessary to reduce the destabilization problems created by HAC. HAC, for example, could be a linear-quadratic-Gaussian (LQG) controller that uses a collection of sensors and actuators. LAC, however, is usually implemented by (active or passive) high-energy-dissipating mechanisms.[4]

High-energy-dissipating mechanisms are usually incorporated into the structure by layers of viscoelastic shear-damping material. In the simplest case, the force-extension characteristic of viscoelastic material can be modeled as a combination of a linear spring and a dash pot, in which the stiffness and the damping are related to the geometry of the dissipating mechanism and the amount of viscoelastic material used. Hence, within the framework of system (1), the parameter $\boldsymbol{x}$ represents, for example, the amount of viscoelastic material at various locations of the structure. A zero $\boldsymbol{x}_i$ would mean that the dissipating mechanism at the corresponding location is not needed, so in many cases it is desirable to find an $\boldsymbol{x}$ with as many zero components as possible (subject to the control design specifications) to obtain a simple design.

Linear state-feedback LAC design is another example that can be easily cast within the framework of system (1). We may require the state-feedback gain to satisfy certain constraints (e.g., on the size of its components or its sparsity pattern) or to find a state-feedback gain that is sparse (so that a small number of sensors/actuators are needed and the controller has a simple topology). This state-feedback approach is particularly useful for the (collocated) rate-feedback design often used for LAC. Specifically, suppose that

$$\dot{z} = Az + B\boldsymbol{u}, \qquad \boldsymbol{u} = Kz$$

where $A \in R^{n\times n}$ and $B \in R^{n\times m}$ are given and $K \in R^{m\times n}$ should be found to achieve, say, sufficient damping. The closed-loop system becomes $\dot{z} = (A + BK)z$ and if $\boldsymbol{x}$ is taken to be the elements of K this problem falls into the framework of system (1). A sparse K represents a simple controller topology because sparsity implies that we need to connect each sensor to only *a few* actuators. Moreover, a zero row (column) in K means that the corresponding actuator (sensor) is not required.

More generally, we can also consider *dynamic* LAC design for the open-loop system

$$\dot{z} = Az + B\boldsymbol{u}, \qquad \boldsymbol{y} = Cz$$

where the controller is parameterized by its state-space system matrices A_c, B_c, C_c, and D_c is given by

$$\dot{z}_c = A_c z_c + B_c \boldsymbol{y}, \qquad \boldsymbol{u} = C_c z_c + D_c \boldsymbol{y}$$

The closed-loop system can be written as

$$\begin{bmatrix} \dot{z} \\ \dot{z}_c \end{bmatrix} = \begin{bmatrix} A + BD_cC & BC_c \\ B_cC & A_c \end{bmatrix} \begin{bmatrix} z \\ z_c \end{bmatrix}$$

which is in the form $\dot{\tilde{z}} = A(\boldsymbol{x})\tilde{z}$ where $\tilde{z} = [z^T \ z_c^T]^T$ and $\boldsymbol{x}$ represents the elements of the controller system matrices A_c, B_c, C_c, and D_c. By requiring sparsity for B_c, C_c, and D_c, we can find designs that require a small number of actuators and sensors.

Another problem that can be formulated within the LAC framework is that of structural design and optimization.[5] In such a case, x can include various parameters such as beamwidths, beam lengths, masses, dampers, etc. The best design, for example, is a structure that supports specified loads at fixed points, achieves acceptable dynamic behavior such as sufficient damping, and at the same time, has the simplest topology or minimum weight.

IV. Linear, Second-Order Cone, and Semidefinite Programming

In this section we briefly introduce linear programs (LPs), second-order cone programs (SOCPs), and semidefinite programs (SDPs) that are families of convex optimization problems that can be efficiently solved (globally) with interior-point methods.[6,7] In later sections, we will see how LAC design can be cast in terms of LPs, SOCPs, or SDPs and hence solved efficiently in practice.

A LP is an optimization problem with linear objective and linear equality and inequality constraints:

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to} && \boldsymbol{f}_i^T\boldsymbol{x} \le g_i, \qquad i = 1, \ldots, J \\ & && A\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{3}$$

where the vector $\boldsymbol{x}$ is the optimization variable and $\boldsymbol{c}, \boldsymbol{f}_i$, g_i, A, and $\boldsymbol{b}$ are problem parameters. Linear programming has been used in a wide variety of fields. In control, for example, Zadeh and Whalen observed in 1962 that certain minimum-time and minimum-fuel optimal control problems could be (numerically) solved by LP.[8] Several high-quality efficient implementations of interior-point LP solvers are available (see, e.g., Refs. 9–11).

A SDP is an optimization problem that has the form

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to} && \boldsymbol{x}_1 F_1 + \cdots + \boldsymbol{x}_m F_m \preceq G \\ & && A\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{4}$$

where F_i and G are symmetric $p \times p$ matrices, and the inequality $\preceq$ denotes matrix inequality, i.e., $X \preceq Y$ means that $Y - X$ is positive semidefinite. The constraint $\boldsymbol{x}_1 F_1 + \cdots + \boldsymbol{x}_m F_m \preceq G$ is called a linear matrix inequality (LMI). Although SDPs look complicated and would appear difficult to solve, new interior-point methods can solve them with great efficiency (see, e.g., Refs. 7 and 12) and several SDP codes* are now widely available.[13–18] The ability to solve SDPs numerically with great efficiency is being applied in several fields, e.g., combinatorial optimization and control.[19] SDP is currently a highly active research area.

A SOCP has the form

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to} && \|F_i\boldsymbol{x} + \boldsymbol{g}_i\| \le \boldsymbol{c}_i^T\boldsymbol{x} + d_i, \qquad i = 1, \ldots, L \\ & && A\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm, i.e., $\|z\| = \sqrt{(z^T z)}$. SOCPs include linear and quadratic programming as special cases, but can also be used to solve a variety of nonlinear, nondifferentiable problems; see, e.g., Ref. 20. Moreover, efficient interior-point software for SOCP is now available.[21,22]

As a final note, it should be mentioned that, among the three different class of optimization problems mentioned, SDP is the most general and includes LP and SOCP as special cases.

V. Eigenvalue-Placement Low-Authority Control Design with Linear and Second-Order Cone Programming

In this section we show that analytic first-order perturbation formulas for eigenvalues of a matrix can be used to design low-authority controllers with linear or second-order cone programming for eigenvalue-placement specifications. As mentioned in Sec. IV, LPs and SOCPs can be solved very efficiently, and therefore this gives an efficient method for LAC design.

A. First-Order Perturbation Formulas for Eigenvalues of a Matrix

A typical problem of the perturbation theory for linear operators is to investigate how the eigenvalues of a linear operator $A \in R^{n\times n}$ change when A is subjected to small perturbation. For example, consider the family of operators $A(\boldsymbol{x}) \in R^{n\times n}$, where $A(0) = A$ and $\boldsymbol{x} \in R^q$ is a parameter that is supposed to be small. A question arises whether the eigenvalues of $A(\boldsymbol{x})$ can be expressed as a power

*Reference 13 is available on-line at http://www-isl.stanford.edu/people/boyd.

series in $\boldsymbol{x}$, i.e., whether they are holomorphic functions of $\boldsymbol{x}$ in the neighborhood of $\boldsymbol{x}=0$.

In Ref. 23 it is shown that, if $A(\boldsymbol{x})$ is m-times continuously differentiable in $\boldsymbol{x}$ on a simply-connected domain $\mathcal{D}\subset R^q$, and the number of eigenvalues $\lambda_i(\boldsymbol{x})$ of $A(\boldsymbol{x})$ corresponding to a Jordan block of size 1 is constant for $\boldsymbol{x}\in\mathcal{D}$, then each $\lambda_i(\boldsymbol{x})$ is also m-times continuously differentiable. Therefore the change of these eigenvalues will be of the same order as the perturbation for small $\|\boldsymbol{x}\|$. Specifically for $m\geq 1$ we have

$$\lambda_i(\boldsymbol{x}) = \lambda_i + \sum_{k=1}^{q}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\boldsymbol{x}_k + o(\|\boldsymbol{x}\|) \tag{6}$$

where $\boldsymbol{u}_i\in C^n$ and $\boldsymbol{w}_i\in C^n$ are the left and the right eigenvectors, respectively, of $A(0)$ corresponding to the eigenvalue $\lambda_i\in C$ and $A_k=\partial A(0)/\partial\boldsymbol{x}_k$ for $k=1,\ldots,q$. Equation (6) gives the first-order expansion formula for the eigenvalues of the perturbed matrix $A(\boldsymbol{x})$.

Remark: If λ_i is a repeated eigenvalue of $A(0)$ corresponding to a Jordan block of size $p_i>1$, $\lambda_i(\boldsymbol{x})$ is no longer given as in Eq. (6). In this case $\lambda_i(\boldsymbol{x})$ is given by a Puiseux series such as

$$\lambda_i(\boldsymbol{x}) = \lambda_i + \sum_{k=1}^{q}\alpha_{ik}\boldsymbol{x}_k^{1/p_i} + \sum_{k=1}^{q}\sum_{j=1}^{q}\beta_{ikj}\boldsymbol{x}_k^{1/p_i}\boldsymbol{x}_j^{1/p_i} + \cdots$$

In other words, the change in the eigenvalue is not of the same order as the perturbation of the matrix for small $\|\boldsymbol{x}\|$.

B. Low-Authority Control Eigenvalue-Placement Design with Linear or Second-Order Cone Programming

Let $\mathcal{D}_i\subset C$ be the desired region for $\lambda_i(\boldsymbol{x})$, the ith eigenvalue of $A(\boldsymbol{x})$. We assume that $\mathcal{D}_i$ is either polyhedral (an intersection of J_i half planes), given by

$$\mathcal{D}_i = \{s\in C\,|\,a_{ij}\mathrm{Re}(s) + b_{ij}\,\mathrm{Im}(s)\leq c_{ij},\ j=1,\ldots,J_i\} \tag{7}$$

where $a_{ij}\in R$, $b_{ij}\in R$, $c_{ij}\in R$, or an intersection of second-order cones given by

$$\mathcal{D}_i = \left\{s\in C\,\left|\,\left\|F_i\begin{bmatrix}\mathrm{Re}(s)\\ \mathrm{Im}(s)\end{bmatrix}+\boldsymbol{g}_i\right\|\leq \boldsymbol{c}_i^T\begin{bmatrix}\mathrm{Re}(s)\\ \mathrm{Im}(s)\end{bmatrix}+d_i\right.\right\} \tag{8}$$

where $F_i\in R^{2\times 2}$, $g_i\in R^2$, $c_i\in R^2$, $d_i\in R$, in which $\mathrm{Re}(s)$ and $\mathrm{Im}(s)$ are the real and the imaginary parts of $s\in C$, respectively (examples of these regions are given below).

Under the LAC assumption, we can drop the $o(\|\boldsymbol{x}\|)$ term in Eq. (6) without significant error, and $\lambda_i(\boldsymbol{x})$ becomes approximately linear in the design variable $\boldsymbol{x}$. From Eq. (6),

$$\mathrm{Re}[\lambda_i(\boldsymbol{x})]\approx \mathrm{Re}(\lambda_i) + \sum_{k=1}^{q}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\boldsymbol{x}_k$$

$$\mathrm{Im}(\lambda_i(\boldsymbol{x}))\approx \mathrm{Im}(\lambda_i) + \sum_{k=1}^{q}\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\boldsymbol{x}_k$$

and therefore to first order $\lambda_i(\boldsymbol{x})\in\mathcal{D}_i$ as defined in Eq. (7) if and only if for $j=1,\ldots,J_i$

$$a_{ij}\left[\mathrm{Re}(\lambda_i) + \sum_{k=1}^{q}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\boldsymbol{x}_k\right] + b_{ij}\left[\mathrm{Im}(\lambda_i) + \sum_{k=1}^{q}\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\boldsymbol{x}_k\right]\leq c_{ij}$$

or equivalently

$$\sum_{k=1}^{q}\left[a_{ij}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right) + b_{ij}\,\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\right]\boldsymbol{x}_k \leq c_{ij} - a_{ij}\,\mathrm{Re}(\lambda_i) - b_{ij}\,\mathrm{Im}(\lambda_i) \tag{9}$$

which is a linear inequality constraint in the variable $\boldsymbol{x}\in R^q$.

Similarly, if we require that $\lambda_i(\boldsymbol{x})$ fall inside the second-order conic region $\mathcal{D}_i$ as in relation (8), to first order we must have

$$\left\|F_i\begin{bmatrix}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right) & \cdots & \mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\\ \mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right) & \cdots & \mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\end{bmatrix}\boldsymbol{x} + F_i\begin{bmatrix}\mathrm{Re}(\lambda_i)\\ \mathrm{Im}(\lambda_i)\end{bmatrix}+\boldsymbol{g}_i\right\|$$

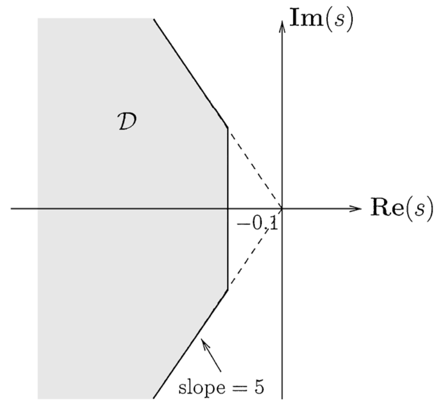
$$\leq \boldsymbol{c}_i^T\begin{bmatrix}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right) & \cdots & \mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\\ \mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right) & \cdots & \mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^*\boldsymbol{u}_i}\right)\end{bmatrix}\boldsymbol{x} + \boldsymbol{c}_i^T\begin{bmatrix}\mathrm{Re}(\lambda_i)\\ \mathrm{Im}(\lambda_i)\end{bmatrix}+d_i \tag{10}$$

which is a second-order cone constraint in $\boldsymbol{x}\in R^q$.

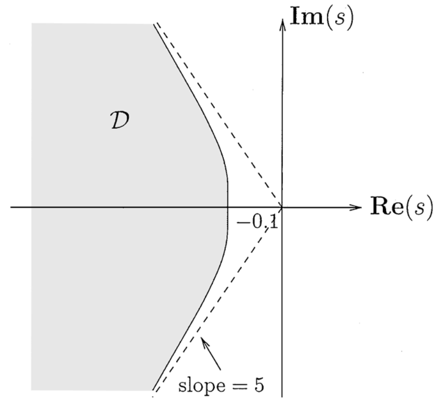
Suitable objectives are usually ones that require $\boldsymbol{x}$ to be in some sense small. These include different norms on $\boldsymbol{x}$ such as $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$, and $\|\boldsymbol{x}\|_\infty$. For example, minimizing $\|\boldsymbol{x}\|_1$ or $\|\boldsymbol{x}\|_\infty$ subject to relation (9) leads to LPs (after slack variables are added), whereas minimizing any of these norms subject to relation (9) or relation (10) leads to SOCPs. Therefore, the LAC eigenvalue placement problem can be easily cast as a LP or SOCP that can be solved very efficiently.

Consider a typical example, which is to place the eigenvalues of system (1) in the shaded region of Fig. 1a (damping or decay rate of at least 0.1, damping ratio of at least 0.2), and the objective is to minimize the sum of the entries of $\boldsymbol{x}$. In this case, for $i=1,\ldots,n$,

$$\mathrm{Re}[\lambda_i(\boldsymbol{x})]\leq -0.1,\qquad \mathrm{Im}[\lambda_i(\boldsymbol{x})]\pm 5\mathrm{Re}[\lambda_i(\boldsymbol{x})]\leq 0$$



a) Polyhedral region



b) Hyperbolic region

Fig. 1 Desired regions for system eigenvalues.

Therefore the optimization problem becomes (to first order)

$$\begin{aligned}
&\text{minimize} \quad \boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_q \\
&\text{subject to} \quad \sum_{k=1}^{q} \operatorname{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \boldsymbol{x}_k \le -0.1 - \operatorname{Re}(\lambda_i) \\
&\sum_{k=1}^{q} \left[\operatorname{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \pm 5\operatorname{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right] \boldsymbol{x}_k \\
&\quad \le -\operatorname{Im}(\lambda_i) \mp 5\operatorname{Re}(\lambda_i) \qquad i = 1, \ldots, n,
\end{aligned}$$

which is a LP in $\boldsymbol{x}$. (Of course, because of conjugate symmetry of the eigenvalues not all of the linear inequality constraints need to be imposed.)

As another example, if the eigenvalues are to be placed in the hyperbolic region $\mathcal{D}$ of Fig. 1b, i.e., $\{s \mid (\sqrt{[\operatorname{Im}(s)^2]} \le -5\operatorname{Re}(s) - 0.5\}$, and the objective is the same as before, according to relation (10) we get the optimization problem

$$\begin{aligned}
&\text{minimize} \quad \boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_q \\
&\text{subject to} \quad \left\| \begin{bmatrix} \operatorname{Im}(\lambda_i) \\ 0 \end{bmatrix} + \begin{bmatrix} \operatorname{Im}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) & \cdots & \operatorname{Im}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \\ 0 & \cdots & 0 \end{bmatrix} \boldsymbol{x} \right\| \\
&\quad \le -5 \left[\operatorname{Re}\left(\frac{\boldsymbol{w}_i^* A_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \cdots \operatorname{Re}\left(\frac{\boldsymbol{w}_i^* A_q \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right] \boldsymbol{x} \\
&\quad - 5\operatorname{Re}(\lambda_i) - 0.5, \qquad i = 1, \ldots, n
\end{aligned}$$

which is a SOCP in $\boldsymbol{x}$.

Note that we can also mix the linear inequality and the second-order cone constraints of relations (9) and (10) with other constraints on $\boldsymbol{x}$. For example, we may require that $0 \le \boldsymbol{x}_i \le \boldsymbol{x}_{i,\max}$ ($\boldsymbol{x}_{i,\max}$ is given) corresponding to, say, physical limitations on the values of $\boldsymbol{x}_i$. As long as these conditions are linear equality, linear inequality, or second-order cone constraints in $\boldsymbol{x}$, they can be easily dealt within an efficient optimization program.

VI. Sparse Low-Authority Control Design

In many cases it is desirable to guarantee performance for system (1) by use of the minimum number of nonzero elements of the vector $\boldsymbol{x}$. For example, each nonzero element could correspond to a sensor, actuator, damper, or structural component, and a *sparse* $\boldsymbol{x}$ (i.e., one with "many" zero elements) would result in a simpler controller, dissipation mechanism, or structure. As another example, $\boldsymbol{x}$ could denote the entries of a full matrix of feedback gains that indicate which sensors should be connected to which actuators. A sparse $\boldsymbol{x}$ then corresponds to a simpler controller topology. In this section we briefly address the problem of computing a sparse $\boldsymbol{x}$ that satisfies one or more of the constraints in Sec. V (or Sec. VIII).

The problem of minimizing the number of nonzero elements of a vector $\boldsymbol{x}$ (subject to some constraints in $\boldsymbol{x}$) arises in many different fields, but unfortunately, except in very special cases, it is a very difficult problem to solve numerically.[24–27] However, a relaxation to this problem gives reasonably sparse solutions while being numerically tractable.[24–27] The method is to minimize the l_1 norm of $\boldsymbol{x}$ instead of minimizing its nonzero entries. The l_1 norm of $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\|_1 = |\boldsymbol{x}_1| + \cdots + |\boldsymbol{x}_q|$, and therefore minimizing $\|\boldsymbol{x}\|_1$ subject to, for example, relation (9) or relation (10) is a LP or SOCP that can be solved very efficiently.

To see why this method is a relaxation to our original problem, let $\|\boldsymbol{x}\|_0$ be the number of nonzero elements of x. Now consider the following optimization problem:

$$\begin{aligned}
&\text{minimize} \quad \|\boldsymbol{x}\|_0 \\
&\text{subject to} \quad \boldsymbol{x} \in \mathcal{C} \\
&\qquad\qquad \|\boldsymbol{x}\|_\infty \le 1
\end{aligned} \tag{11}$$

where $\mathcal{C}$ is some compact (convex) subset of R^n. (Note that by scaling variables, without loss of generality, we can assume that $\|\boldsymbol{x}\|_\infty \le 1$.) Optimization problem (11) can be cast as the mixed optimization problem

$$\begin{aligned}
&\text{minimize} \quad z_1 + z_2 \cdots + z_q \\
&\text{subject to} \quad \boldsymbol{x} \in \mathcal{C} \\
&\qquad |\boldsymbol{x}_i| \le z_i \\
&\qquad z_i \in \{0, 1\}, \qquad i = 1, \ldots, q
\end{aligned}$$

The Boolean constraint $z_i \in \{0, 1\}$ is what makes the above optimization problem numerically intractable (NP hard), because, roughly speaking, to solve this problem exactly, one should solve an exponential number of feasibility problems corresponding to the 2^q possibilities for the vector z. Now if we *relax* the Boolean constraint $z_i \in \{0, 1\}$ by the convex constraint $0 \le z_i \le 1$ we get the convex optimization problem

$$\begin{aligned}
&\text{minimize} \quad z_1 + z_2 \cdots + z_q \\
&\text{subject to} \quad \boldsymbol{x} \in \mathcal{C} \\
&\qquad |\boldsymbol{x}_i| \le z_i \\
&\qquad 0 \le z_i \le 1, \qquad i = 1, \ldots, q
\end{aligned}$$

which is the same as

$$\begin{aligned}
&\text{minimize} \quad \|\boldsymbol{x}\|_1 \\
&\text{subject to} \quad \boldsymbol{x} \in \mathcal{C} \\
&\qquad\qquad \|\boldsymbol{x}\|_\infty \le 1
\end{aligned} \tag{12}$$

Hence, we have just shown that problem (12) is a natural relaxation to problem (11). (When we do not have the constraint $\|\boldsymbol{x}\|_\infty \le 1$, the natural relaxation will involve a weighted l_1 norm.)

The l_1 norm relaxation method, although generally suboptimal, results in a significant reduction in computational complexity and enables us to deal with sparse problems that are extremely difficult, if not impossible, to solve exactly. Moreover, as demonstrated in Sec. IX, this method tends to give acceptable sparse solutions in practice: solving problem (12) typically gives a vector $\boldsymbol{x}$ with many entries that are exactly equal to zero.

However, if we insist on finding the optimum $\boldsymbol{x}$, we need to enumerate all possible sparsity patterns of $\boldsymbol{x}$ and check them for feasibility (i.e., if there exists an $\boldsymbol{x}$ with the given sparsity pattern that satisfy the constraints). Among all feasible sparsity patterns of $\boldsymbol{x}$, the one with the least number of nonzero elements minimizes $\|\boldsymbol{x}\|_0$. Since $\boldsymbol{x}$ has q components and each component is either zero or nonzero, the total number of sparsity patterns of $\boldsymbol{x}$ is 2^q. Therefore, in principle, by solving at most 2^q feasibility problems it is possible to find an $\boldsymbol{x}$ that minimizes $\|\boldsymbol{x}\|_0$. But 2^q could be very large for even relatively small values of q, and, as a result, finding the optimum $\boldsymbol{x}$ could be cumbersome. Good heuristics as to how and in what order to check the different sparsity patterns of $\boldsymbol{x}$ usually greatly reduce the necessary number of feasibility problems we need to solve. For example, experience in Sec. IX has indicated that one heuristic is to use the solution to the l_1 relaxation problem as a basis to decide what sparsity patterns should be checked first (it is more likely for relatively large components of the relaxed solution to be nonzero).

VII. Robust Low-Authority Control Design

In this section we address the problem of robust LAC design, i.e., a LAC design with guaranteed (closed-loop) system performance subject to uncertainties or variations in the system model. We show that it is possible to solve the robust LAC design problem by using LP and SOCP. Therefore, by combining the methods of this section and that of Sec. VI, we can handle low-authority controller design, actuator/sensor placement, and robustness at the same time. Robust actuator/sensor placement and robust controller design are usually performed in two separate stages (and hence nonoptimally) because it is numerically intractable to do otherwise (see, e.g., Ref. 28 for a thorough overview of robust actuator and damper placement for structural control). As demonstrated in Sec. IX, it is yet another numerical advantage of LAC design that it is possible to handle both of these problems in one step very efficiently.

We consider two different approaches for modeling the system uncertainty and show how to design a robust LAC in each case. The first approach is to consider a parametric uncertainty, and the second approach is to model the uncertainty by a finite number of possible system models. The uncertainty is assumed to be time invariant in both cases.

A. Robust Low-Authority Control Design for Systems Subject to Small Parametric Uncertainties

As a generalization to the setup of Sec.I, we assume that the dynamics of the (closed-loop) system can be described as

$$\dot{z} = A(\boldsymbol{x}, \boldsymbol{\delta})z \tag{13}$$

where $\boldsymbol{x} \in R^q$ is the design parameter (as before), and $\boldsymbol{\delta} \in R^r$ represents the model uncertainty satisfying

$$-\boldsymbol{\delta}_{i,\max} \leq \boldsymbol{\delta}_i \leq \boldsymbol{\delta}_{i,\max} \tag{14}$$

for $i = 1, \dots, r$ in which $\delta_{i,\max}$ is given. We assume that the low-authority assumption holds and $\boldsymbol{\delta}$ is small so that the eigenvalues of $A(\boldsymbol{x}, \boldsymbol{\delta})$ can be well approximated by (first-order) perturbation formulas. The goal is to find $\boldsymbol{x}$ such that for all possible values of δ, the eigenvalues of Eq. (13) are in some desired region of the complex plane. Let $\mathcal{D}_i \subset C$ be the desired region for $\lambda_i(\boldsymbol{x}, \boldsymbol{\delta})$, the ith eigenvalue of $A(\boldsymbol{x}, \boldsymbol{\delta})$. We assume that $\mathcal{D}_i$ is polyhedral, as in Eq. (7).

When the Farkas lemma (see, e.g., Ref. 5) is used, it can be shown that (to first order) $\lambda_i(\boldsymbol{x}, \boldsymbol{\delta}) \in \mathcal{D}_i$ for all $\boldsymbol{\delta}$ satisfying relation (14) if and only if there exists $\boldsymbol{\tau}^{(1)}, \boldsymbol{\tau}^{(2)} \in R^r$ such that

$$\boldsymbol{\tau}_l^{(1)} \geq 0, \qquad \boldsymbol{\tau}_l^{(2)} \geq 0$$

$$\boldsymbol{\tau}_l^{(1)} - \boldsymbol{\tau}_l^{(2)} = a_{ij}\,\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* \bar{A}_l \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) + b_{ij}\,\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* \bar{A}_l \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)$$

$$\sum_{k=1}^{q}\left[a_{ij}\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) + b_{ij}\,\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right]\boldsymbol{x}_k$$

$$+ \sum_{l=1}^{r}\left[\boldsymbol{\tau}_l^{(1)} + \boldsymbol{\tau}_l^{(2)}\right]\boldsymbol{\delta}_{l,\max} \leq c_{ij} - a_{ij}\,\mathrm{Re}(\lambda_i) - b_{ij}\,\mathrm{Im}(\lambda_i) \tag{15}$$

for $l = 1, \dots, r$ and $j = 1, \dots, J_i$, which is a set of linear equality and inequality constraints in $\boldsymbol{x}$, $\boldsymbol{\tau}^{(1)}$, and $\boldsymbol{\tau}^{(2)}$. Hence, by minimizing $\|\boldsymbol{x}\|_1$ subject to relation (15), for example, it is possible to design robust and sparse LACs for eigenvalue-placement specifications subject to bounded parametric uncertainties in the system model by solving LPs.

Note that, if similar methods are used, it is possible to cast robust LAC design as a LP or SOCP for cases in which $\mathcal{D}_i$ is described as in Eq. (8) and/or $\boldsymbol{\delta}$ is bound to lie in an ellipsoid. Ellipsoidal (confidence) regions for $\boldsymbol{\delta}$ may come from a statistical study of the uncertainties in the system. For example, suppose that it is known that the uncertainty $\boldsymbol{\delta}$ lies in the ellipsoid $\{\boldsymbol{\delta} \mid \|F\boldsymbol{\delta} + \boldsymbol{g}\| \leq 1\}$ where $F \in R^{r\times r}$ (full rank) and $\boldsymbol{g} \in R^r$ are known. It is easy to verify that, to first order, the ith eigenvalue of Eq. (13) lies in $\mathcal{D}_i$ as defined in Eq. (7) if and only if

$$\sum_{k=1}^{q}\left[a_{ij}\,\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) + b_{ij}\,\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* A_k \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right]\boldsymbol{x}_k$$

$$\leq c_{ij} - a_{ij}\,\mathrm{Re}(\lambda_i) - b_{ij}\,\mathrm{Im}(\lambda_i) + \left\|F^{-T}\boldsymbol{d}_{ij}\right\| + \boldsymbol{d}_{ij}^T F^{-1}\boldsymbol{g}$$

for $j = 1, \dots, J_i$, where

$$\boldsymbol{d}_{ij} = a_{ij}\left[\mathrm{Re}\left(\frac{\boldsymbol{w}_i^* \bar{A}_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \cdots \mathrm{Re}\left(\frac{\boldsymbol{w}_i^* \bar{A}_r \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right]^T$$

$$+ b_{ij}\left[\mathrm{Im}\left(\frac{\boldsymbol{w}_i^* \bar{A}_1 \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right) \cdots \mathrm{Im}\left(\frac{\boldsymbol{w}_i^* \bar{A}_r \boldsymbol{u}_i}{\boldsymbol{w}_i^* \boldsymbol{u}_i}\right)\right]^T$$

Again, these are a set of linear inequalities in $\boldsymbol{x}$ and can therefore be handled by solving LPs.

B. Robust Low-Authority Control Design for Systems with Multiple Models

Here we consider a multiple model approach to robust LAC design. This approach relies on the fact that it is possible to model uncertainty or plant variation adequately by a finite number of system models:

$$\dot{z} = A^{(l)}(\boldsymbol{x})z, \qquad l = 1, \dots, \nu \tag{16}$$

For a robust LAC design within this framework, the goal is to find $\boldsymbol{x}$ such that the eigenvalues of each of the system models of Eq. (16) is in some desired region of the complex plane. This can be easily done by requiring the eigenvalue-placement specifications to hold for each of the models.

For example, if the desired region $\mathcal{D}_i$ of the ith eigenvalue is given as in Eq. (7), then by using perturbation formulas from Sec. V.A, we require that

$$\sum_{k=1}^{q}\left\{a_{ij}\,\mathrm{Re}\left[\frac{\boldsymbol{w}_i^{(l)*} A_k^{(l)} \boldsymbol{u}_i^{(l)}}{\boldsymbol{w}_i^{(l)*}\boldsymbol{u}_i^{(l)}}\right] + b_{ij}\,\mathrm{Im}\left[\frac{\boldsymbol{w}_i^{(l)*} A_k^{(l)} \boldsymbol{u}_i^{(l)}}{\boldsymbol{w}_i^{(l)*}\boldsymbol{u}_i^{(l)}}\right]\right\}\boldsymbol{x}_k$$

$$\leq c_{ij} - a_{ij}\,\mathrm{Re}\left[\lambda_i^{(l)}\right] - b_{ij}\,\mathrm{Im}\left[\lambda_i^{(l)}\right] \tag{17}$$

for $l = 1, \dots, \nu$, where $\lambda_i^{(l)}$, $\boldsymbol{u}_i^{(l)}$, and $\boldsymbol{w}_i^{(l)}$ are the ith eigenvalue, right eigenvector, and left eigenvector of $A^{(l)}(0)$, respectively. Therefore, in the robust case, eigenvalue-placement specifications can still be described as LPs that are just ν times larger. Hence, robust LAC design with a multiple model approach can be easily handled as before.

VIII. Extension: Low-Authority Control Design Based on Lyapunov Theory

In this section we show how Lyapunov theory and SDPs can be used to design low-authority controllers for more advanced design objectives than the eigenvalue specifications in Secs. V. These design objectives can be combined to get, for example, a desired eigenvalue location for the system while providing a bound on output energy or the $\mathcal{L}_2$ gain. Or, by combining the results of Secs. VI and VII with those presented here, it is possible to perform robust actuator/sensor placement or controller structure design that are optimum to first order for a variety of control objectives.

The idea of LAC design by using Lyapunov theory is similar to the idea of LAC design for eigenvalue placement of Sec. V. Again, the perturbations to the open-loop system are assumed to be small, and hence linear approximations can be used to predict accurately the behavior of the closed-loop system for controller design.

Briefly, to explain the linear approximation in this case, suppose that the Lyapunov function $V(z) = z^T P z$, $P \succ 0$ proves a level of performance for some property of the unperturbed or open-loop system (2). Then, under the low-authority assumption, since the perturbations $\boldsymbol{x}_i$ to the open-loop system are small, the Lyapunov function $\hat{V}(z) = z^T(P + \delta P)z$, $P + \delta P \succ 0$ with small δP is a Lyapunov function candidate for the same property of closed-loop system (1). Therefore, as a first-order approximation, we can neglect second-order (cross) terms such as $\boldsymbol{x}_i \delta P$ in the (bilinear) matrix inequality conditions that are equivalent to $\hat{V}$ being a Lyapunov function, proving a (better) level of performance for the closed-loop system. Hence the matrix inequalities become jointly linear in $\boldsymbol{x}$ and δP and therefore can be easily handled by solving SDPs.

In this section we illustrate this method for two different design specifications, but it should be noted that the method is quite powerful and can also be applied to handle many other design specifications.

A. Bound on Output Energy

Consider the (closed-loop) linear dynamical system with output

$$\dot{z} = A(\boldsymbol{x})z, \qquad \boldsymbol{y} = C(\boldsymbol{x})z, \qquad z(0) = z_0 \tag{18}$$

The goal is to design $\boldsymbol{x}$ to moderately reduce the output energy $\int_0^\infty \boldsymbol{y}^T\boldsymbol{y}\,\mathrm{d}t$ of closed-loop system (18) from that of the unperturbed or open-loop system [i.e., system (18) with $\boldsymbol{x} = 0$].

The output energy of the open-loop system is bounded by $z(0)^T P z(0)$ for any $P \succ 0$ satisfying (see, e.g., Ref. 19)

$$A(0)^T P + P A(0) + C(0)^T C(0) \preceq 0 \tag{19}$$

[If the inequality in this equation is replaced by equality, $z(0)^T Pz(0)$ gives the exact output energy.] The output energy of closed-loop system (18) is bounded by $z(0)^T(P + \delta P)z(0)$ if there exists δP such that $P + \delta P \succ 0$ and

$$A(\boldsymbol{x})^T(P+\delta P) + (P+\delta P)A(\boldsymbol{x}) + C(\boldsymbol{x})^T C(\boldsymbol{x}) \preceq 0 \qquad (20)$$

Under the low-authority assumption it is reasonable to assume that δP and $\boldsymbol{x}_i$ are small and their product is, to first order, negligible. Hence, by expanding $A(\boldsymbol{x})$ and $C(\boldsymbol{x})$ in relation (20) to their first-order (Taylor) approximation and neglecting the second-order terms such as $\boldsymbol{x}_i\delta P$, we get

$$A(0)^T P + PA(0) + C(0)^T C(0) + A(0)^T\delta P + \delta PA(0) + \sum_{k=1}^{q}\boldsymbol{x}_k\left[A_k^T P + PA_k + C(0)^T C_k + C_k^T C(0)\right] \preceq 0 \qquad (21)$$

where $A_k \triangleq \partial A(0)/\partial \boldsymbol{x}_k$ and $C_k \triangleq \partial C(0)/\partial \boldsymbol{x}_k$. Relation (21) is a LMI in the variables $\delta P \in R^{n\times n}$ and $\boldsymbol{x} \in R^q$, although we do not write out the LMI explicitly in the standard form $\boldsymbol{x}_1 F_1 + \cdots + \boldsymbol{x}_m F_m \preceq G$ of Sec. IV (leaving LMIs in condensed form, in addition to saving notation, may lead to more efficient computation). When the constraint $P + \delta P \succeq 0$ is added (and $\|\delta P\| \le 0.2\|P\|I$, for example, is constrained to ensure that the first-order approximations are accurate), a first-order condition for an output energy of $z(0)^T(P + \delta P)z(0)$ for the closed-loop system becomes

$$P + \delta P \succeq 0, \qquad \begin{bmatrix} 0.2P & \delta P \\ \delta P & 0.2P \end{bmatrix} \succeq 0$$

$$A(0)^T P + PA(0) + C(0)^T C(0) + A(0)^T\delta P + \delta PA(0) + \sum_{k=1}^{q}\boldsymbol{x}_k\left[A_k^T P + PA_k + C(0)^T C_k + C_k^T C(0)\right] \preceq 0 \qquad (22)$$

where P is any positive-definite matrix satisfying relation (19) [e.g., the unique solution to the Lyapunov equation $A(0)^T P + PA(0) + C(0)^T C(0) = 0$]. By adding (linear) constraints such as $z(0)^T(P + \delta P)z(0) \le \epsilon$ or $\mathrm{Tr}(P + \delta P) \le \eta$ that require the output energy to be smaller than some prescribed level and by minimizing, for example, $\|\boldsymbol{x}\|_1$, we can solve LAC design for output energy specifications by using SDP.

B. $\mathcal{L}_2$ Gain

Consider the (closed-loop) linear dynamical system with input and output

$$\dot{z} = A(\boldsymbol{x})z + B(\boldsymbol{x})\boldsymbol{w}, \qquad \boldsymbol{y} = C(\boldsymbol{x})z + D(\boldsymbol{x})\boldsymbol{w} \qquad (23)$$

Suppose that the induced $\mathcal{L}_2$ gain from input $\boldsymbol{w}$ to output $\boldsymbol{y}$ of the open-loop system, i.e., system (23) with $\boldsymbol{x} = 0$, is less than γ so that[19]

$$\begin{bmatrix} A(0)^T P + PA(0) + C(0)^T C(0) & PB(0) + C(0)^T D(0) \\ B(0)^T P + D(0)^T C(0) & -\gamma^2 I + D(0)^T D(0) \end{bmatrix} \preceq 0 \qquad (24)$$

Then, with reasoning similar to that of Sec. VIII.A, to first order, the induced $\mathcal{L}_2$ gain from input $\boldsymbol{w}$ to output $\boldsymbol{y}$ of the closed-loop system is less than $\gamma^2 + \delta(\gamma^2)$ if

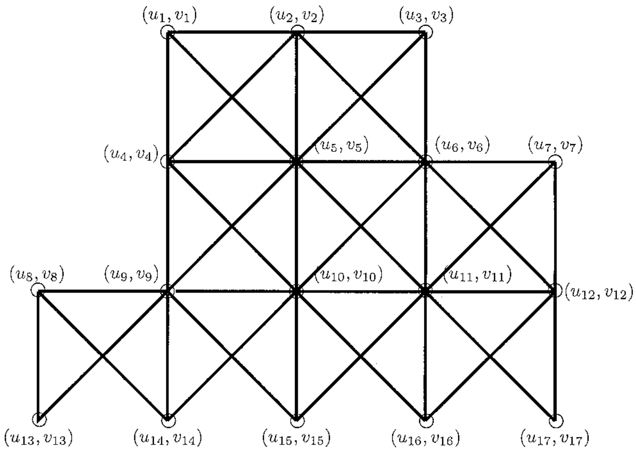


Fig. 2 Truss structure consists of 39 bars (stiffness and damping) and 17 nodes (masses).

where $A_k \triangleq \partial A(0)/\partial \boldsymbol{x}_k$, $B_k \triangleq \partial B(0)/\partial \boldsymbol{x}_k$, $C_k \triangleq \partial C(0)/\partial \boldsymbol{x}_k$, and $D_k \triangleq \partial D(0)/\partial \boldsymbol{x}_k$. Relation (25) is a LMI in the variables $\delta P = \delta P^T$, $\boldsymbol{x}$, and $\delta(\gamma^2)$.

Note that the method of this section works for any P that proves some level of performance for the open-loop system [e.g., any P that satisfies relation (19) or relation (24)]. This observation highlights a potential weakness of this method because it is not clear which P should be used in, for example, relation (21) or relation (25). However, we conjecture that it does not make much difference which P is chosen because the perturbations are assumed to be small and the P can be adjusted by the free variable δP. Our experience indicates that the P with the smallest condition number or the one that minimizes $\log\det P^{-1}$ seems to work well in practice.

As a final remark, it should be noted that the different LAC design constraints in the preceding sections can be mixed freely. Because these constraints were either linear inequalities, second-order cone constraints, or LMIs, a SDP solver (e.g., Ref. 15) can be used to compute the design parameter $\boldsymbol{x}$ very efficiently in practice.

For example, $\boldsymbol{x}$ can be a vector of feedback gains of different collocated sensor/actuator pairs, and for the closed-loop system we may require a certain minimum amount of damping and damping ratio in the eigenvalues (see Sec. V.B) and a bound on the level of the induced $\mathcal{L}_2$ norm from an input to an output (see Sec. VIII.B), while the ith feedback gain is absolutely bounded by $\boldsymbol{x}_{i,\max}$ ($-\boldsymbol{x}_{i,\max} \le \boldsymbol{x}_i \le \boldsymbol{x}_{i,\max}$). By minimizing $\|\boldsymbol{x}\|_1$ subject to these design specifications, it is hoped that we will obtain a sparse $\boldsymbol{x}$ and therefore many of the sensor/actuator pairs will not be needed (see Sec. VI). Section IX presents several examples that illustrate this design procedure.

IX. Example: Low-Authority Control Design for 39-bar Truss Structure

The purpose of this section is to design low-authority controllers for the truss structure shown in Fig. 2. The structure consists of 39 bars with stiffness and damping connecting 17 masses at the nodes. The dynamics of the structure are written as $\dot{z} = Az$, where $A \in R^{64\times 64}$ and the state variable z consists of (a linear combination of) the horizontal and the vertical displacements and rates

$$P + \delta P \succeq 0, \qquad \begin{bmatrix} 0.2P & \delta P \\ \delta P & 0.2P \end{bmatrix} \succeq 0$$

$$\begin{bmatrix} \left\{\begin{array}{l} A(0)^T P + PA(0) + C(0)^T C(0) + A(0)^T\delta P + \delta PA(0) \\ \quad + \sum_{k=1}^{q}\boldsymbol{x}_k\left(A_k^T P + PA_k\right) + \sum_{k=1}^{q}\boldsymbol{x}_k\left[C(0)^T C_k + C_k^T C(0)\right] \end{array}\right\} & \left\{\begin{array}{l} (P+\delta P)B(0) + C(0)^T D(0) \\ \quad + \sum_{k=1}^{q}\boldsymbol{x}_k\left[PB_k^T + C_k^T D(0) + C(0)^T D_k\right] \end{array}\right\} \\ \left\{\begin{array}{l} B(0)^T(P+\delta P) + D(0)^T C(0) \\ \quad + \sum_{k=1}^{q}\boldsymbol{x}_k\left[B_k P + D(0)^T C_k + D_k^T C(0)\right] \end{array}\right\} & \left\{\begin{array}{l} -[\gamma^2 + \delta(\gamma^2)]I + D(0)^T D(0) \\ \quad + \sum_{k=1}^{q}\boldsymbol{x}_k\left[D_k^T D(0) + D(0)D_k\right] \end{array}\right\} \end{bmatrix} \preceq 0 \qquad (25)$$

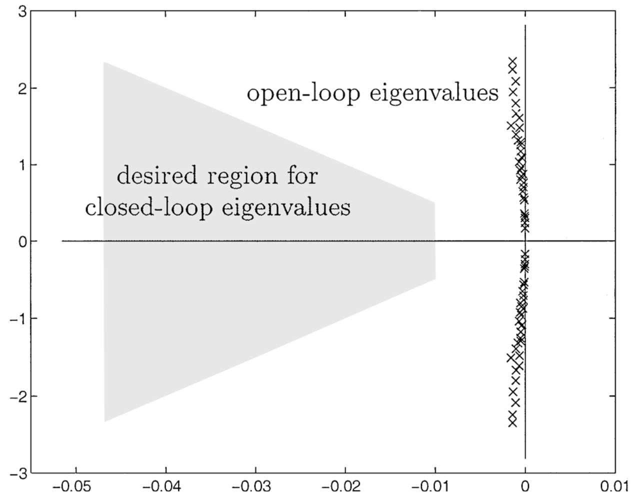


Fig. 3 Open-loop eigenvalues of structure and the desired region for closed-loop eigenvalues.

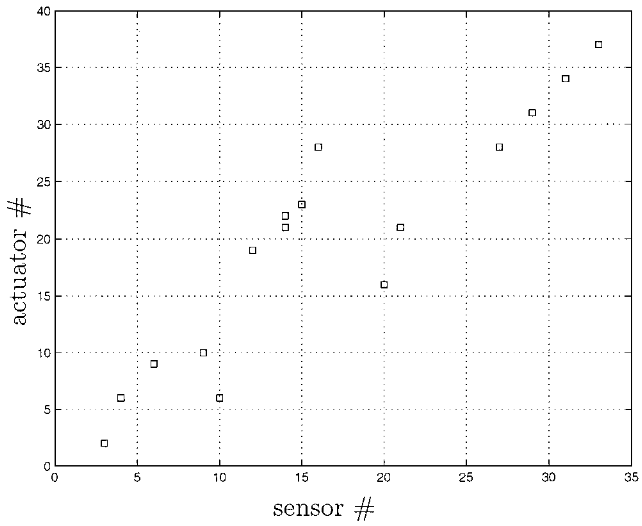


Fig. 6 Sparsity pattern of the feedback gain matrix. A nonzero feedback gain from that sensor to actuator is indicated by a box.

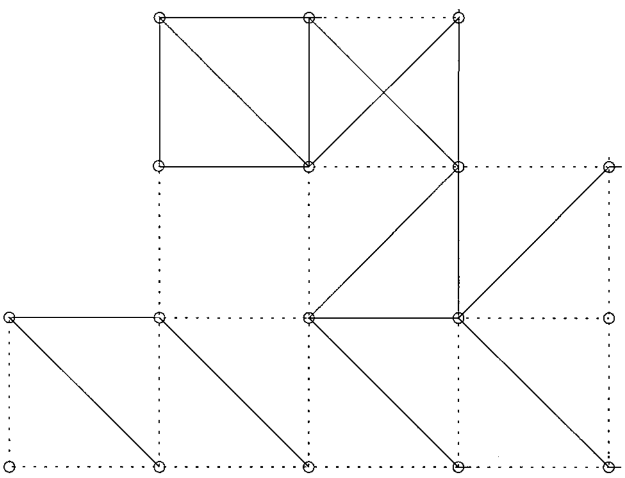


Fig. 4 Location of dampers for the LAC design. A solid line between two nodes corresponds to a nonzero damper between those two nodes.

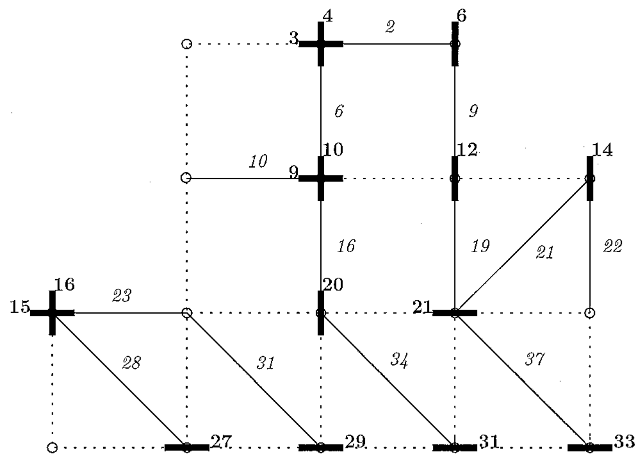


Fig. 7 Sensor and actuator locations. A solid line between two nodes corresponds to an actuator between those two nodes. A vertical or horizontal line crossing a node corresponds to a vertical or horizontal rate sensor at that node. The actuator numbers are italicized and the sensor numbers are boldfaced.

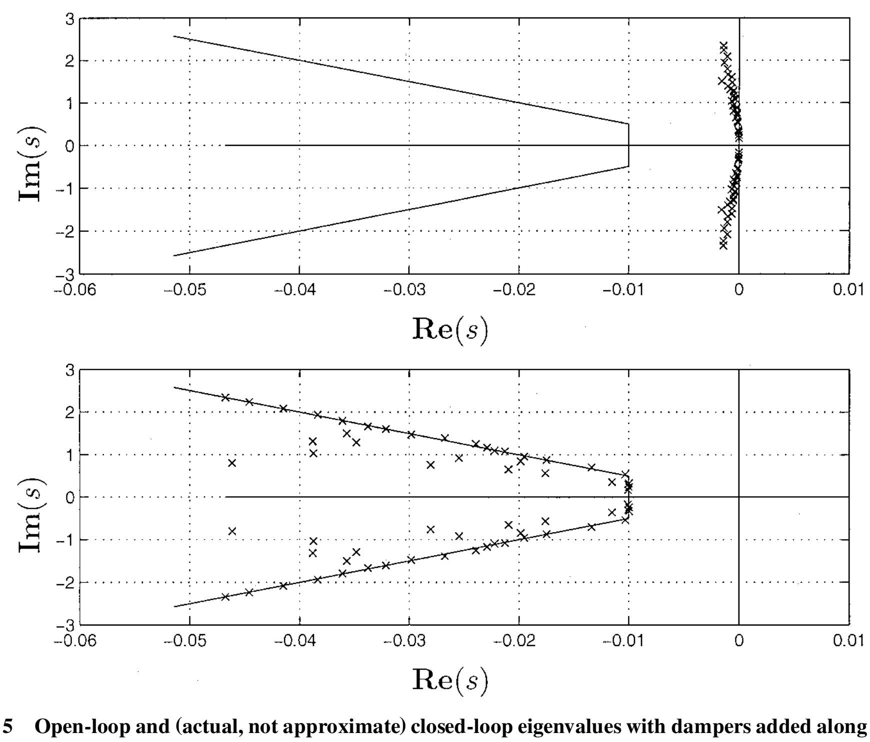


Fig. 5 Open-loop and (actual, not approximate) closed-loop eigenvalues with dampers added along bars.

of displacements of each mass, u_i, v_i, $\dot{u}_i$, and $\dot{v}_i$, respectively, for $i = 1, \ldots, 17$.

The goal is to design a controller that achieves an overall damping or decay rate of at least 0.01 and a damping ratio of at least 0.02. The open-loop eigenvalues and the desired region for the closed-loop eigenvalues of the system are shown in Fig. 3. We assume that the low-authority assumption holds and use the method of Sec. V.B to design controllers that achieve the required decay rate and damping ratio. The validity of the low-authority assumption will be verified after each design.

A. Low-Authority Control by Use of Dampers Along Bars

We first consider the case in which we can place a damper of size b_i along each bar to achieve the design specifications. The closed-loop system dynamics are now written as $\dot{z} = A(x)z$, where the design variables are the size of each of the dampers, $x = [b_1\ b_2\ \cdots\ b_{39}]^T$ and $A(0) = A$. In this case, $A(x)$ is affine in x.

It is desirable to find a design in which many of the dampings are zero. To achieve such a design we minimize the l_1 norm of x subject to the eigenvalue placement constraints. (Note that the number of sparsity patterns of x is $2^{39} \approx 10^{12}$ and an exhaustive

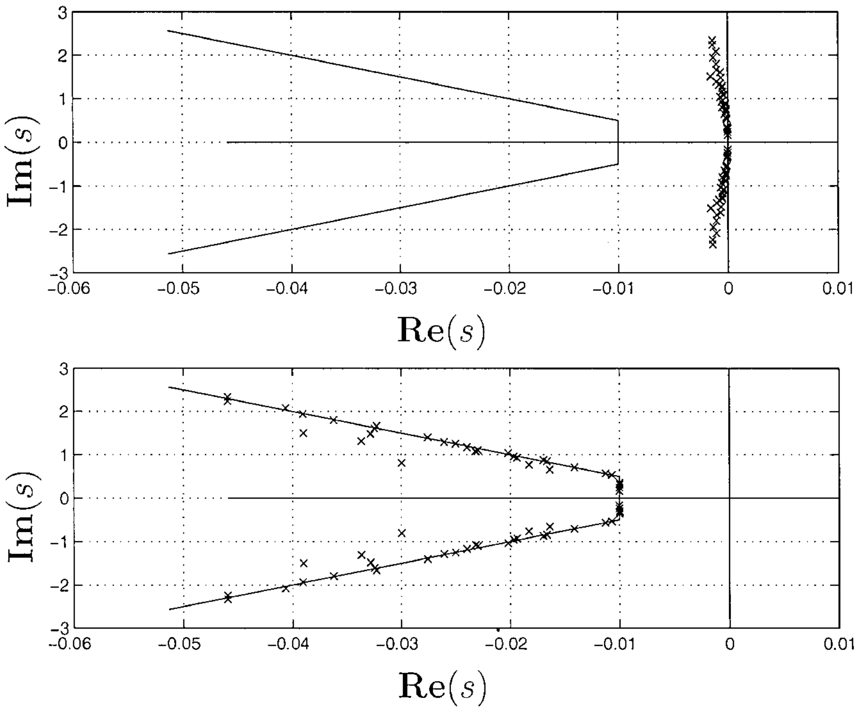


Fig. 8 Open-loop and (actual, not approximate) closed-loop eigenvalues with rate sensors at nodes and force actuators along bars.

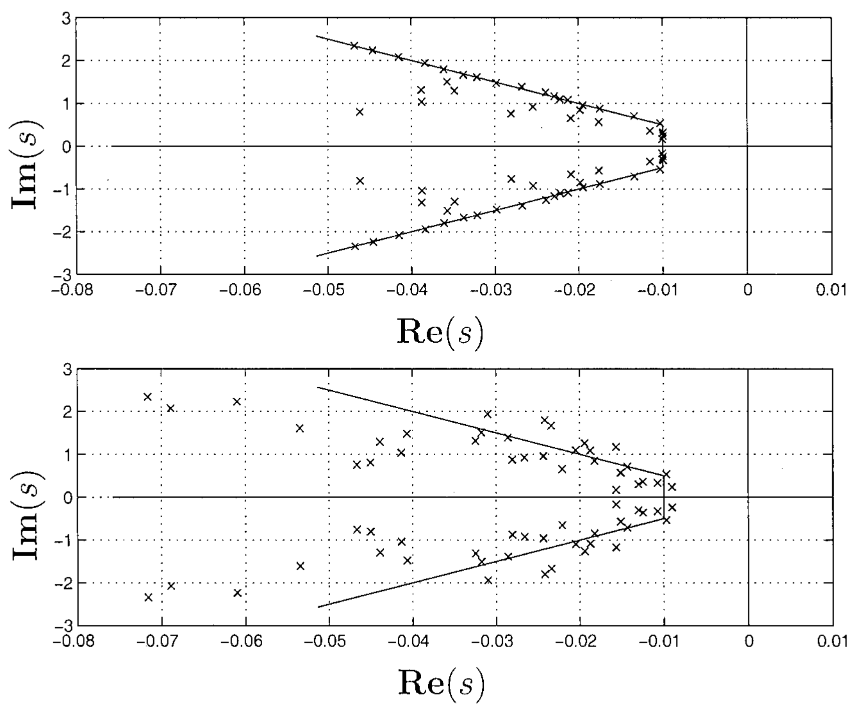


Fig. 9 Closed-loop eigenvalues for model 1 (top) and model 2 (bottom) with dampers designed for model 1.

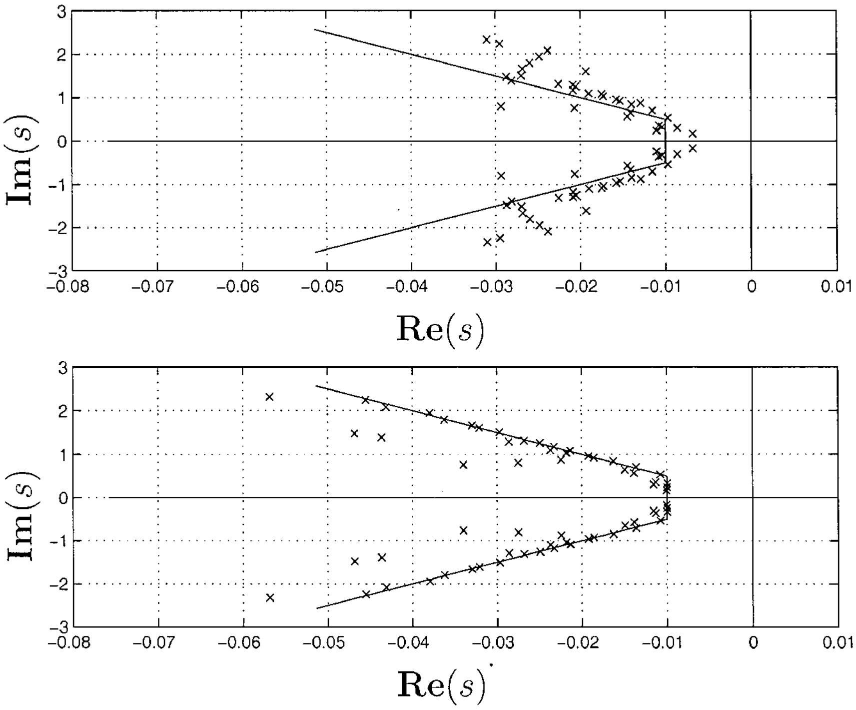


Fig. 10 Closed-loop eigenvalues for model 1 (top) and model 2 (bottom) with dampers designed for model 2.

search method for computing the optimum $\|\boldsymbol{x}\|_0$ is impractical.) The resulting LP that must be solved to find the damping design consists of 39 variables and 64 linear inequality constraints. The solution for this problem resulted in 22 out of the 39 possible dampers being zero. (This takes several seconds with the LP solver PCx* on a typical personal computer.) The location of the nonzero dampers are shown in Fig. 4. A solid line between two nodes in this figure corresponds to a nonzero damper between those two nodes. The figure shows that, in this case, most of the dampers are on the diagonals of the truss structure. In this case we get $\sum_i b_i = 1.73$ and $\max_i b_i = 0.27$, which are measures of the total amount of damping material that must be added to the structure and to a single strut.

To verify the low-authority assumption, Fig. 5 shows a plot of the actual (not first-order approximate) eigenvalues of the closed-loop system. All closed-loop poles satisfy the requirements, or are very close to the boundary, which clearly shows that the low-authority assumption is valid in this case.

B. Low-Authority Control with Rate Sensors at Each Node, Force Actuator Along Each Bar

A more sophisticated design approach is to use active damping. In this case we assume that a rate sensor can be placed at each node (measuring $\dot{u}_i$ and $\dot{v}_i$) and a force actuator can be placed along each bar. We consider an extremely flexible control architecture that allows each sensor to be connected to each actuator by means of a feedback gain that must be determined. The dynamics of the closed-loop system are written as $\dot{z} = A(\boldsymbol{x})z$ such that the vector of design variables $\boldsymbol{x} \in R^{1326}$ represents the elements of the 34×39 matrix of feedback gains from each sensor to each actuator. In this case, $A(\boldsymbol{x})$ is again affine in $\boldsymbol{x}$.

The goal is to achieve the eigenvalue placement design specifications with a small number of actuators/sensors and a simple controller topology. This objective is accomplished by minimizing the l_1 norm of $\boldsymbol{x}$ subject to the eigenvalue placement specifications, which is a LP with 1326 variables and 64 linear inequality constraints. (This takes approximately a minute with PCx on a typical personal computer.)

The sparsity pattern of the resulting feedback gain matrix is given in Fig. 6. Again, the solution is very sparse and only 16 out of 1326 possible feedback gains are nonzero ($\sum_i |\boldsymbol{x}_i| = 3.44$ and

*PCx can be downloaded from the World Wide Web at URL http://www-c.mcs.anl.gov/home/otc/Library/PCx/.

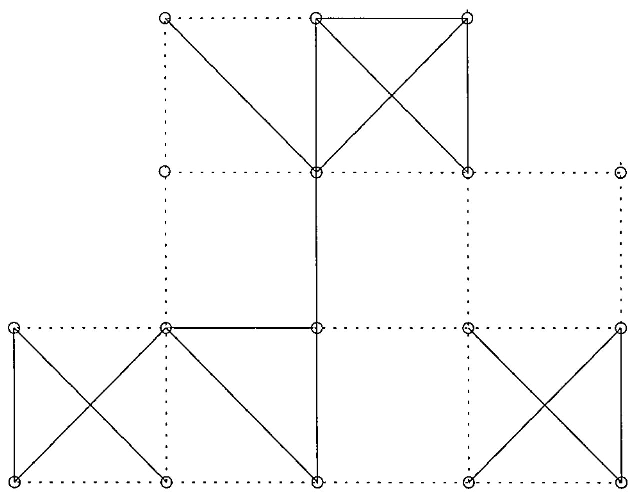


Fig. 11 Location of dampers for design based on model 2. A solid line between two nodes corresponds to a nonzero damper between those two nodes.

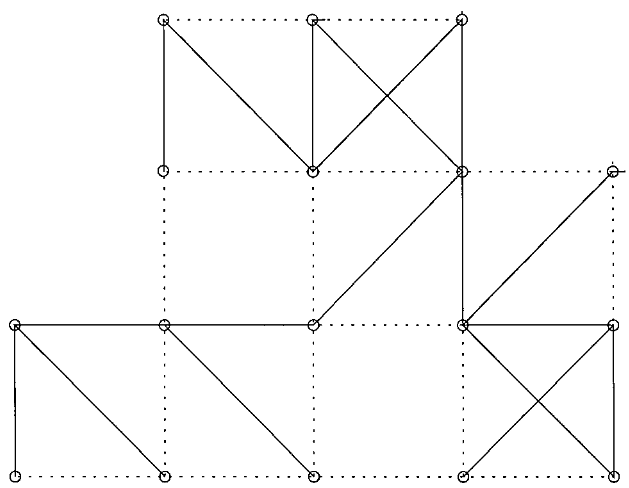


Fig. 12 Location of dempers for robust design. A solid line between two nodes corresponds to a nonzero damper between those two nodes.

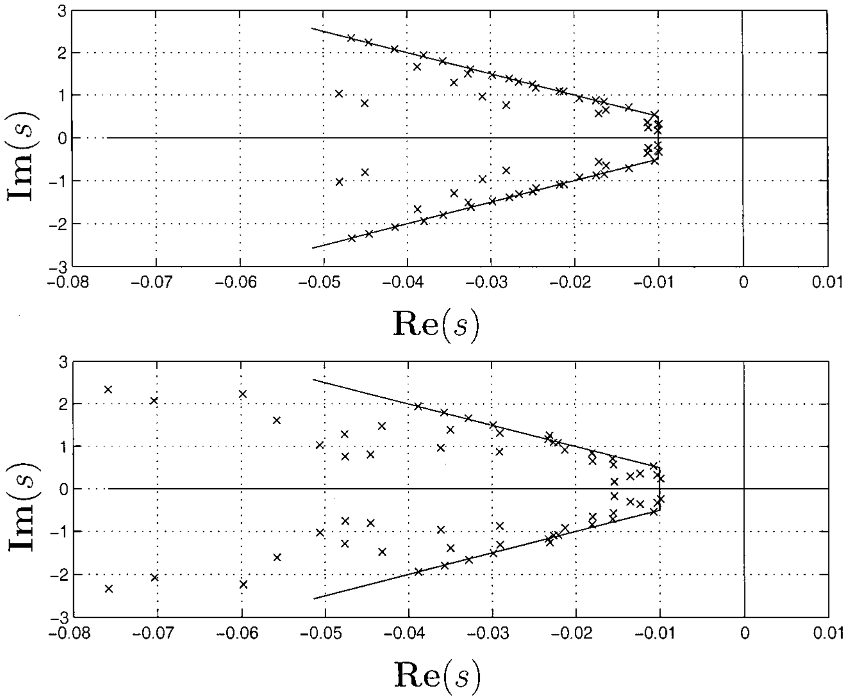


Fig. 13 Closed-loop eigenvalues for model 1 (top) and model 2 (bottom) with robust design.

$\max_i |x_i| = 0.56$). Of course, actuators (sensors) that are not connected to a sensor (actuator) can be eliminated. For the solution given here, only 13 (out of 39) actuators, and 15 (out of 34) sensors are required. Figure 7 shows the location of these actuators and sensors.

By examining Figs. 6 and 7, we can see that the controller is collocated rate feedback, in the sense that there is no feedback path from a sensor to an actuator that does not have the sensor attached to it. It is interesting to note that actuators 6, 21, and 28 use two sensors, whereas all other actuators use only one sensor. Also, all sensors are connected to only one actuator except for 14, which is connected to two actuators, 21 and 22.

Thus, by considering a problem with a very general feedback matrix, the optimization has succeeded in simultaneously performing the sensor/actuator placement problem and the feedback control design. The open-loop and (actual) closed-loop eigenvalues for this design are shown in Fig. 8. Again, the closed-loop poles meet or exceed the design specifications, which verifies the low-authority assumption in this design approach.

C. Robust Low-Authority Control Example for Truss

A key problem with sensor actuator placement problems is the sensitivity of the optimization to the particular system model used.[28] To address this problem, we design a robust LAC based on a multiple model approach, as discussed in Sec. VII.B. We assume that there are two possible models for the truss: model 1 is the same truss considered in the preceding examples, and model 2 is the same truss but with the node masses doubled at the third level and halved at the second level. As before, the goal is to find the amount of damping required along each bar for robustly placing the eigenvalues of the system in the desired region in Fig. 3. Once again, we solve this problem by optimizing the l_1 norm of $\boldsymbol{x}$ to find the amount of damping required on each strut.

Robustness is a concern in this problem, because if the dampers are designed solely for model 1, then the eigenvalues of the closed-loop system corresponding to model 2 will not satisfy the eigenvalue-placement specifications. This result is shown in Fig. 9. The design problem that uses model 1 was discussed in Sec. IX.A, and the nonzero damping locations are shown in Fig. 4.

Figure 10 shows the actual closed-loop eigenvalues for both models for a design using model 2. As before, if the dampers are solely designed for model 2, then the closed-loop system corresponding to model 1 violates the eigenvalue-placement specifications. The nonzero damping locations selected using model 2 are shown in Fig. 11. A comparison of Figs. 4 and 11 indicates that there are some similarities in the best locations for the dampers based on these two models (primarily in the first and third level), but the two solutions are very different in the second level.

As discussed, we can resolve this difficulty using a robust LAC design that is based on both model 1 and model 2. The nonzero damping locations for the robust solution are given in Fig. 12. This design is interesting because, as the figure shows, it, combines many of the unique placement features of the two nonrobust designs. However, for the design based on model 1, $\sum_i b_i = 1.73$, for the design based on model 2, $\sum_i b_i = 1.52$, and for the robust design $\sum_i b_i = 1.82$. Thus the robustness is achieved with only slightly more damping in the structure. The actual closed-loop eigenvalue locations corresponding to each model are shown in Fig. 13. Clearly, the eigenvalue-placement specifications hold for both models.

X. Conclusions

In this paper it was shown that linear, second-order cone, and semidefinite programs, which are subclasses of convex problems that can be efficiently solved numerically, can be used to solve very complex low-authority control problems. Design objectives include eigenvalue placement, robustness, sparsity of feedback gains or actuator/sensor placement (using the l_1 relaxation heuristic), output energy, and $\mathcal{L}_2$ gain (using Lyapunov methods). These design objectives can be freely mixed to design, for example, the location of sensors and actuators and feedback gains of a controller that achieves a desired eigenvalue location for the closed-loop system while providing a bound on the $\mathcal{L}_2$ gain subject to parametric uncertainties in the plant model. Several numerical examples were also given that demonstrate the effectiveness of the design approach.

As a final remark, note that it is possible to design medium-authority or high-authority controllers by iteratively using the low-authority design method of this paper. Starting from the initial (open-loop) system, the idea is to design better and better controllers by slowly improving the design objective (e.g., given a fixed architecture controller we could iteratively design for lower values of the $\mathcal{L}_2$ gain). Because the design objectives in consecutive problems are close, then at each step, the linearized matrix inequalities can be used to accurately design a controller that improves on the previous one by solving a convex optimization problem. This path-following (homotopy) method offers a new method for solving (locally) bilinear matrix inequalities in control.[29]

Acknowledgments

This research was supported by the U.S. Air Force under Grant F49620-97-1-0459,the U.S. Air Force Office of Scientific Research under Grant F49620-95-1-0318,the U.S. National Science Foundation under Grants ECS-9222391 and EEC-9420565, and the Multidisciplinary University Research Initiative under Grant F49620-95-1-0525.

References

[1]Wykes, J. H., "Structural Dynamic Stability Augmentation and Gust Alleviation of Flexible Aircraft," AIAA Paper 68-1067, Oct. 1968.

[2]Aubrun, J. N., "Theory of Control of Structures by Low-Authority Controllers," *Journal of Guidance and Control*, Vol. 3, No. 5, 1980, pp. 444–451.

[3]Williams, T., "Transmission Zeros and High-Authority/Low-Authority Control of Flexible Space Structures," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 1, 1994, pp. 170–174.

[4]Sun, C. T., and Lu, Y. P., *Vibration Damping of Structural Elements*, Prentice–Hall, Englewood Cliffs, NJ, 1995.

[5]Vanderbei, R. J., *Linear Programming: Foundations and Extensions*, Kluwer Academic, Norwell, MA, 1997.

[6]Nesterov, Yu., and Nemirovsky, A., *Interior-Point Polynomial Methods in Convex Programming*, Studies in Applied Mathematics Series, Vol. 13, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[7]Vandenberghe, L., and Boyd, S., "A Primal-Dual Potential Reduction Method for Problems Involving Matrix Inequalities," *Mathematical Programming*, Vol. 69, No. 1, 1995, pp. 205–236.

[8]Zadeh, L. A., and Whalen, B. H., "On Optimal Control and Linear Programming," *Institute of Radio Engineers Transactions on Automatic Control*, Vol. 7, No. 1, 1962, pp. 45, 46.

[9]Vanderbei, R. J., "LOQO User's Manual," Dept. of Civil Engineering and Operations Research, TR SOL 92-05, Princeton University, Princeton, NJ, 1992.

[10]Zhang, Y., "User's Guide to LIPSOL: A Matlab Toolkit for Linear Programming Interior-Point Solvers," Mathematics and Statistics Dept., Univ. of Maryland, Baltimore, Oct. 1994.

[11]Czyzyk, J., Mehrotra, S., and Wright, S. J., "PCx User Guide," Optimization Technology Center, Argonne National Lab., Argonne, IL, March 1997.

[12]Alizadeh, F., Haeberly, J.-P. A., and Overton, M. L., "Primal-Dual Interior-Point Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results," *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 746–768.

[13]Vandenberghe, L., and Boyd, S., "SP: Software for Semidefinite Programming. User's Guide, Beta Version," Stanford Univ., Stanford, CA, Oct. 1994.

[14]Fujisawa, K., and Kojima, M., "SDPA (Semidefinite Programming Algorithm) User's Manual," Dept. of Mathematical and Computing Sciences, TR B-308, Tokyo Inst. of Technology, Tokyo, Japan, 1995.

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

[16]Gahinet, P., and Nemirovskii, A., "LMI Lab: A Package for Manipulating and Solving LMIs," French National Inst. for Research in Science and Control, Rocquencourt, France, 1993.

[17]Alizadeh, F., Haeberly, J. P., Nayakkankuppam, M. V., and Overton, M. L., "SDPPACK User's Guide, Version 0.8 Beta," New York Univ., New York, June 1997.

[18]Borchers, B., "CSDP, A C Library for Semidefinite Programming," New Mexico Tech. Inst., New Mexico, March 1997.

[19]Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Studies in Applied Mathematics Series, Vol. 15, Society for Industrial and Applied Mathematics, Philadelphia, PA, June 1994.

[20]Lobo, M. S., Vandenberghe, L., Boyd, S., and Lebret, H., "Applications of Second-Order Cone Programming," *Linear Algebra and Applications*, Vol. 284, Nos. 1–3, 1998, pp. 193–228.

[21]Lobo, M. S., Vandenberghe, L., and Boyd, S., "SOCP: Software for Second-Order Cone Programming," Information Systems Laboratory, Stanford Univ., Stanford, CA, 1997.

[22]Alizadeh, F., Haeberly, J. P., Nayakkankuppam, M. V., Overton, M. L., and Schmieta, S., "SDPPACK User's Guide, Version 0.9 Beta," New York Univ., New York, June 1997.

[23]Kato, T., *A Short Introduction to Perturbation Theory for Linear Operators*, Springer–Verlag, New York, 1982.

[24]Chen, S. S., Donoho, D., and Saunders, M. A., "Atomic Decomposition of Basis Pursuit," Dept. of Statistics, TR 479, Stanford Univ., Stanford, CA, Feb. 1996.

[25]Rao, B. D., and Gorodnitsky, I. F., "Affine Scaling Transformation Based Methods for Computing Low Complexity Sparse Solutions," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Inst. of Electrical and Electronics Engineers, New York, 1996, Vol. 3, pp. 1783–1786.

[26]Harikumar, G., and Bresler, Y., "A New Algorithm for Computing Sparse Solutions to Linear Inverse Problems," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Inst. of Electrical and Electronics Engineers, New York, 1996, Vol. 3, pp. 1331–1334.

[27]Mesbahi, M., and Papavassilopoulos, G. P., "On the Rank Minimization Problem over a Positive Semidefinite Linear Matrix Inequality," *IEEE Transactions on Automatic Control*, Vol. 42, No. 2, 1997, pp. 239–243.

[28]Anderson, E. H., and Hagood, N. W., "Robust Placement of Actuators and Dampers for Structural Control," Space Engineering Research Center, TR SERC #14-93, Massachusetts Inst. of Technology, Cambridge, MA, Oct. 1993.

[29]Hassibi, A., How, J. P., and Boyd, S. P., "A Path-Following Method for Solving BMI Problems in Control," *Proceedings of American Control Conference*, American Automatic Control Council, Evanston, IL, June 1999, pp. 1385–1389.