# Low-Authority Controller Design via Convex Optimization[1]

Arash Hassibi[2] Jonathan How Stephen Boyd

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

## 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* using 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.

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 very large order problems can be solved (globally) in practice. Another advantage of this formulation is that it can handle a very wide variety of specifications and objectives beyond standard eigenvalue-placement. Typical objectives for 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 $\ell_1$ 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.

The paper is organized as follows. The next section poses the problem statement, which is followed by a section that presents typical applications of LAC. Section 3 discusses the first order perturbation formulas for the matrix eigenvalues, and how the design problem can be posed within convex optimization framework. Section 4 discusses the sparsity of the solution, which is important for the control architecture studies. Section 5 addresses robust LAC design, *i.e.*, a LAC design that guarantees performance subject to uncertainties or variations in the plant model. Section 6 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, Section 7 demonstrates the application of the methods on a few example problems.

## 1 Problem statement

We consider the linear time-invariant system

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

where $z(t) \in \mathbf{R}^n$ is the state, $x \in \mathbf{R}^q$ is a (design) parameter, and $A(x) \in \mathbf{R}^{n\times n}$ is differentiable at $x = 0$. The goal is to find $x$ 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 $x$ so that the eigenvalues of system (1) are only slightly different from the eigenvalues of the *unperturbed* system

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

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

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

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

## 2 Applications of LAC

A key control design methodology for flexible systems with many elastic modes follows the two-level architecture presented in [1, 3]. This architecture consists of a wide-band, low-authority control (LAC) and a narrow-band, high-authority control (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, due to spillover, the HAC can destabilize modes not included in the design model, which are usually at

[1]Research supported by Air Force (under F49620-97-1-0459), AFOSR (under F49620-95-1-0318), and NSF (under ECS-9222391 and EEC-9420565).

[2]Contact author. Email: arash@isl.stanford.edu

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 using a collection of sensors and actuators. LAC, however, is usually implemented using high-energy-dissipating mechanisms such as layers of viscoelastic shear damping material. In this case, the parameter $x$ in (1) could represent the amount of viscoelastic material at various locations of the structure.

Linear state-feedback LAC design is another simple example that can be easily cast in the framework (1). In this case, the parameter $x$ consists of the elements of the state-feedback gain matrix. We may require the state-feedback gain to satisfy certain constraints (*e.g.*, on the size of its components or its sparsity pattern), or find a state-feedback gain that is sparse. This state-feedback approach is particularly useful for the (collocated) rate-feedback design often used for LAC. A sparse feedback gain matrix represents a simple controller topology since it implies that we only need to connect each sensor to a *few* actuators. Moreover, a zero row (column) means that the corresponding actuator (sensor) is not required.

More generally, we can also consider *dynamic* LAC design where the controller is parameterized by its state-space system matrices. In this case, $x$ represents the elements of these matrices. By requiring sparsity, we can find designs that require a small number of actuators and sensors.

Another problem that can be formulated in the LAC framework is that of structural design and optimization. In such a case, $x$ can include various parameters such as beam widths, 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.

## 3 Eigenvalue-placement LAC design using 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 using linear programming (LP) or second-order cone programming (SOCP) for eigenvalue-placement specifications. LPs and SOCPs are convex optimization problems and can be solved very efficiently, both in theory and practice (see, *e.g.*, [4, 5, 6, 7]).

### 3.1 First order perturbation formulas for eigenvalues of a matrix

Consider the family of operators $A(x) \in \mathbf{R}^{n\times n}$ where $A(0) = A$ and $x \in \mathbf{R}^q$ is a parameter supposed to be small. A question arises whether the eigenvalues of $A(x)$ can be expressed as a power series in $x$, *i.e.*, whether they are holomorphic functions of $x$ in the neighborhood of $x = 0$.

In [8] it is shown that if $A(x)$ is continuously differentiable in $x$ on a simply-connected domain $\mathcal{D} \subset \mathbf{R}^q$, and the number of eigenvalues $\lambda_i(x)$ of $A(x)$ corresponding to a Jordan block of size 1 is constant for $x \in \mathcal{D}$, then each $\lambda_i(x)$ is also continuously differentiable. Therefore, the change of these eigenvalues will be of the same order of magnitude as the perturbation for small $\|x\|$. Specifically we have

$$\lambda_i(x) = \lambda_i + \sum_{k=1}^{q} \left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right) x_k + o(\|x\|), \tag{3}$$

where $u_i \in \mathbf{C}^n$, $w_i \in \mathbf{C}^n$ are the left and right eigenvectors of $A(0)$ corresponding to the eigenvalue $\lambda_i \in \mathbf{C}$, and $A_k = \partial A(0)/\partial x_k$ for $k = 1, \ldots, q$. Equation (3) gives the first order expansion formula for the eigenvalues of the perturbed matrix $A(x)$.

### 3.2 LAC eigenvalue-placement design using linear or second-order cone programming

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

$$\mathcal{D}_i = \{\ s \in \mathbf{C} \mid a_{ij}\, \mathbf{Re}(s) + b_{ij}\, \mathbf{Im}(s) \le c_{ij},\ j = 1, \ldots, J_i\ \}, \tag{4}$$

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

$$\mathcal{D}_i = \{\ s \in \mathbf{C} \mid \left\| F_i \begin{bmatrix} \mathbf{Re}(s) \\ \mathbf{Im}(s) \end{bmatrix} + g_i \right\| \le c_i^T \begin{bmatrix} \mathbf{Re}(s) \\ \mathbf{Im}(s) \end{bmatrix} + d_i\ \}, \tag{5}$$

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

Under the low-authority control assumption, we can drop the $o(\|x\|)$ term in equation (3) without significant error, and $\lambda_i(x)$ becomes approximately linear in the design variable $x$, and therefore, to first order $\lambda_i(x) \in \mathcal{D}_i$ as defined in (4) if and only if (after some simple algebraic manipulations)

$$\begin{aligned} \textstyle\sum_{k=1}^{q} \left(a_{ij}\, \mathbf{Re}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right) + b_{ij}\, \mathbf{Im}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right)\right) x_k \le \\ c_{ij} - a_{ij}\, \mathbf{Re}\left(\lambda_i\right) - b_{ij}\, \mathbf{Im}\left(\lambda_i\right), \end{aligned} \tag{6}$$

for $j = 1, \ldots, J_i$, which are linear inequality constraints in $x$.

Similarly, if we require that $\lambda_i(x)$ fall inside the second-order conic region $\mathcal{D}_i$ as in (5), to first order we must have

$$\left\| F_i H_i x + F_i \begin{bmatrix} \mathbf{Re}(\lambda_i) \\ \mathbf{Im}(\lambda_i) \end{bmatrix} + g_i \right\| \le c_i^T H_i x + c_i^T \begin{bmatrix} \mathbf{Re}(\lambda_i) \\ \mathbf{Im}(\lambda_i) \end{bmatrix} + d_i, \tag{7}$$

where

$$H_i = \begin{bmatrix} \mathbf{Re}\left(\frac{w_i^* A_1 u_i}{w_i^* u_i}\right) & \cdots & \mathbf{Re}\left(\frac{w_i^* A_q u_i}{w_i^* u_i}\right) \\ \mathbf{Im}\left(\frac{w_i^* A_1 u_i}{w_i^* u_i}\right) & \cdots & \mathbf{Im}\left(\frac{w_i^* A_q u_i}{w_i^* u_i}\right) \end{bmatrix}.$$

Note that (7) is a second-order cone constraint in $x$.

Suitable objectives are usually ones that require $x$ to be in some sense "small". These include different norms on $x$ such as $\|x\|_1$, $\|x\|_2$, and $\|x\|_\infty$. For example, minimizing $\|x\|_1$ or $\|x\|_\infty$ subject to (6) leads to LPs (after adding slack variables), while minimizing any of these norms subject to (6) or (7) leads to SOCPs. Therefore, the LAC eigenvalue-placement problem can be easily cast as an LP or SOCP which can then be solved very efficiently.

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

$$\mathbf{Re}\left(\lambda_i(x)\right) \leq -0.1, \quad \mathbf{Im}\left(\lambda_i(x)\right) \pm 5\,\mathbf{Re}\left(\lambda_i(x)\right) \leq 0.$$

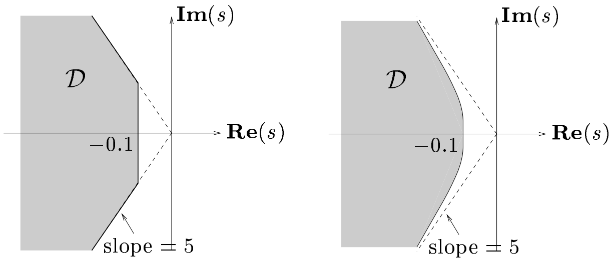
Therefore, the optimization problem becomes (to first order)

$$\begin{array}{ll} \text{minimize} & x_1 + x_2 + \cdots + x_q \\ \text{subject to} & \sum_{k=1}^{q} \mathbf{Re}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right) x_k \leq -0.1 - \mathbf{Re}\left(\lambda_i\right), \\ & \sum_{k=1}^{q} \left(\mathbf{Im}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right) \pm 5\,\mathbf{Re}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right)\right) x_k \\ & \qquad \leq -\mathbf{Im}\left(\lambda_i\right) \mp 5\,\mathbf{Re}\left(\lambda_i\right), \quad i = 1, \ldots, n, \end{array}$$

which is an LP in $x$.

As another example, if the eigenvalues are to be placed in the hyperbolic region $\mathcal{D}$ of Figure 1(b), *i.e.*, $\{s \mid \sqrt{\mathbf{Im}(s)^2} \leq -5\,\mathbf{Re}(s) - 0.5\}$, and the objective is the same as before, using (7) we get an SOCP (minimizing a linear function over second-order cone constraints).

Note that we can also mix the linear inequality and second-order cone constraints (6) and (7) with other constraints on $x$. For example, we may require that $0 \leq x_i \leq x_{i,\max}$ ($x_{i,\max}$ is given) corresponding to, say, physical limitations on the values of $x_i$. As long as these conditions are linear equality, linear inequality, or second-order cone constraints in $x$, they can be easily dealt within an efficient optimization program.




(a) Polyhedral $\mathcal{D}$. (b) Hyperbolic $\mathcal{D}$.

**Figure 1:** Desired regions for system eigenvalues.

## 4 Sparse LAC design

In many cases it is desirable to guarantee performance for system (1) using the minimum number of nonzero elements of the vector $x$. For example, each nonzero element could correspond to a sensor, actuator, damper, or structural component, and a *sparse* $x$ (*i.e.*, one with "many" zero elements) would result in a simpler controller, dissipation mechanism, or structure. As another example, $x$ could denote the entries of a full matrix of feedback gains that indicates which sensors should be connected to which actuators. A sparse $x$ then corresponds to a simpler controller topology.

The problem of minimizing the number of nonzero elements of a vector $x$ (subject to some constraints in $x$) arises in many different fields, but unfortunately, except in very special cases, it is a very difficult problem to solve numerically. However, a relaxation to this problem gives reasonably sparse solutions while being numerically tractable. The method is to minimize the $\ell_1$ norm of $x$ instead of minimizing its nonzero entries. The $\ell_1$ norm of $x$ is defined as $\|x\|_1 = |x_1| + \cdots + |x_q|$, and therefore, minimizing $\|x\|_1$ subject to, for example, (6) or (7) is an LP or SOCP that can be solved very efficiently.

The $\ell_1$ norm relaxation method usually tends to give acceptable sparse solutions. However, if we insist on finding the $x$ with the least number of nonzero elements, we need to enumerate all possible sparsity patterns of $x$ (there are $2^q$ of them) and check them for feasibility (*i.e.*, if there exists an $x$ with the given sparsity pattern that satisfy the constraints). Among all feasible sparsity patterns of $x$, the one with the least number of nonzero elements is the solution.

## 5 Robust LAC 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 using LP and SOCP. Therefore, by combining the methods of this section and that of §4, we can handle low-authority controller design, actuator/sensor placement, and robustness at the same time.

We will consider two different approaches for modeling the system uncertainty. 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.

### 5.1 Robust LAC design for systems subject to "small" parametric uncertainties

As a generalization to the setup of §1, we assume that the dynamics of the (closed-loop) system can be described as

$$\dot{z} = A(x, \delta) z \tag{8}$$

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

$$-\delta_{i,\max} \leq \delta_i \leq \delta_{i,\max} \tag{9}$$

for $i = 1, \ldots, r$ in which $\delta_{i,\max}$ is given. We assume that the low-authority assumption holds and $\delta$ is "small" so that the eigenvalues of $A(x, \delta)$ can be well-approximated using (first order) perturbation formulas. The goal is to find $x$ such that for all possible values of $\delta$, the eigenvalues of (8) are in some desired region of the complex plane. Let $\mathcal{D}_i \subset \mathbf{C}$ be the desired region for $\lambda_i(x, \delta)$, the $i$th eigenvalue of $A(x, \delta)$. We assume that $\mathcal{D}_i$ is polyhedral as in (4).

Using the Farkas Lemma, it can be shown that (to first order) $\lambda_i(x, \delta) \in \mathcal{D}_i$ for all $\delta$ satisfying (9) if and only if there exists $\tau^{(1)}, \tau^{(2)} \in \mathbf{R}^r$ such that

$$\begin{array}{l} \tau_l^{(1)} \geq 0, \quad \tau_l^{(2)} \geq 0, \\ \tau_l^{(1)} - \tau_l^{(2)} = a_{ij}\,\mathbf{Re}\left(\frac{w_i^* \bar{A}_l u_i}{w_i^* u_i}\right) + b_{ij}\,\mathbf{Im}\left(\frac{w_i^* \bar{A}_l u_i}{w_i^* u_i}\right), \\ \sum_{k=1}^{q} \left(a_{ij}\,\mathbf{Re}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right) + b_{ij}\,\mathbf{Im}\left(\frac{w_i^* A_k u_i}{w_i^* u_i}\right)\right) x_k + \\ \quad \sum_{l=1}^{r} \left(\tau_l^{(1)} + \tau_l^{(2)}\right) \delta_{l,\max} \leq c_{ij} - a_{ij}\,\mathbf{Re}\left(\lambda_i\right) - b_{ij}\,\mathbf{Im}\left(\lambda_i\right), \end{array} \tag{10}$$

for $l = 1, \ldots, r$ and $j = 1, \ldots, J_i$, which is a set of linear equality and inequality constraints in $x$, $\tau^{(1)}$, and $\tau^{(2)}$. Hence, by minimizing $\|x\|_1$ subject to (10) for example, it is possible to design robust and sparse LACs for eigenvalue-placement specifications subject to bounded parametric uncertainties by solving LPs.

Note that using similar methods, it is possible to cast robust LAC design as an LP or SOCP for cases in which $\mathcal{D}_i$ is described as in (5), and/or $\delta$ is bound to lie in an ellipsoid.

### 5.2 Robust LAC design for systems with multiple models

Here we consider a multiple model approach to robust LAC design. In this approach, we suppose that it is possible to adequately model uncertainty or plant variation by a finite number of system models given by

$$\dot{z} = A^{(l)}(x)z, \quad l = 1, \ldots, \nu. \tag{11}$$

For robust LAC design in this framework, the goal is to find $x$ such that the eigenvalues of each of the system models (11) are 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. Therefore, in the robust case, eigenvalue-placement specifications can still be described as LPs that are just $\nu$ times larger.

## 6 Extension: LAC design based on Lyapunov theory

In this section we show how Lyapunov theory can be used to design low-authority controllers. Lyapunov methods are very powerful and enable us to formulate design objectives *beyond* eigenvalue-placement specifications in terms of semidefinite programs (SDPs), which can then be solved very efficiently (see, *e.g.*, [9, 10, 11]). These design objectives can be combined to get, for example, a desired eigenvalue location for the system while providing a bound on output energy, $\mathcal{L}_2$ gain, *etc.*

Here, the method is illustrated on the *output energy* objective, but it should be noted that the method is quite powerful, and can also be applied to handle many other design specifications [12].

### 6.1 Bound on output energy

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

$$\dot{z} = A(x)z, \quad y = C(x)z, \quad z(0) = z_0. \tag{12}$$

The goal is to design $x$ to "moderately" reduce the output energy $\int_0^\infty y^T y \, dt$ of the closed-loop system (12) from that of the unperturbed or open-loop system (*i.e.*, system (12) with $x = 0$).

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.*, [13])

$$A(0)^T P + PA(0) + C(0)^T C(0) \preceq 0. \tag{13}$$

(If the inequality in this equation is replaced by equality, $z(0)^T P z(0)$ gives the exact output energy). The output energy of the closed-loop system (12) is bounded by $z(0)^T (P+\delta P) z(0)$ if there exists $\delta P$ such that $P+\delta P \succ 0$ and

$$A(x)^T (P + \delta P) + (P + \delta P)A(x) + C(x)^T C(x) \preceq 0. \tag{14}$$

Under the low-authority assumption it is reasonable to assume that $\delta P$ and $x_i$ are "small" and their product is to first order negligible. Hence, by expanding $A(x)$ and $C(x)$ in (14) to their first order (Taylor) approximation, and neglecting the second order terms such as $x_i \delta P$ we get

$$\begin{aligned} &A(0)^T P + PA(0) + C(0)^T C(0) + A(0)^T \delta P + \delta P A(0) + \\ &\quad \textstyle\sum_{k=1}^{q} x_k \left( A_k^T P + P A_k + C(0)^T C_k + C_k^T C(0) \right) \preceq 0, \end{aligned} \tag{15}$$

where $A_k \triangleq \partial A(0)/\partial x_k$, and $C_k \triangleq \partial C(0)/\partial x_k$. (15) is a linear matrix inequality (LMI) in the variables $\delta P \in \mathbf{R}^{n \times n}$ and $x \in \mathbf{R}^q$. By adding the constraint $P + \delta P \succeq 0$ (and constraining $\|\delta P\| \leq 0.2\|P\| I$ for example to ensure that the first order approximations are accurate), a first order condition for an output energy of no more than $z(0)^T (P + \delta P) z(0)$ for the closed-loop system becomes

$$(15), \quad P + \delta P \succeq 0, \quad \begin{bmatrix} 0.2P & \delta P \\ \delta P & 0.2P \end{bmatrix} \succeq 0, \tag{16}$$

where $P$ is any positive definite matrix satisfying (13). By adding (linear) constraints such as $z(0)^T (P + \delta P) z(0) \leq \epsilon$, $\mathbf{Tr}(P + \delta P) \leq \eta$, *etc.*, that require the output energy to be smaller than some prescribed level, and by minimizing for example $\|x\|_1$, LAC design for output energy specifications can be solved using SDP.

Note that the method of this section works for any $P$ that satisfies (13). This observation highlights a potential weakness of this method because it is not clear which $P$ should be used. However, we conjecture that it does not make much difference which $P$ is chosen, because $P$ can be adjusted using the free variable $\delta P$. Our experience indicates that, selecting either the $P$ which is the unique solution to the Lyapunov equation $A(0)^T P + PA(0) + C(0)^T C(0) = 0$, the $P$ with smallest condition number satisfying (13), or the $P$ that minimizes $\log\det P^{-1}$ subject to (13) works well in practice.

## 7 Example: LAC design for 39-bar truss structure

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

The goal is to design a controller that achieves an overall damping of at least 0.01 and a damping ratio of at least 0.02. The open-loop eigenvalues and the desired region $\mathcal{D}$ for the closed-loop eigenvalues of the system are shown in Figure 3. We will assume that the low-authority assumption holds and use the method of §3.2 to design controllers that achieve the required damping and damping ratio. The validity of the low-authority assumption is verified after each design.

### 7.1 LAC using 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 also desirable to find a design in which many of the dampings are zero. To achieve such a design,

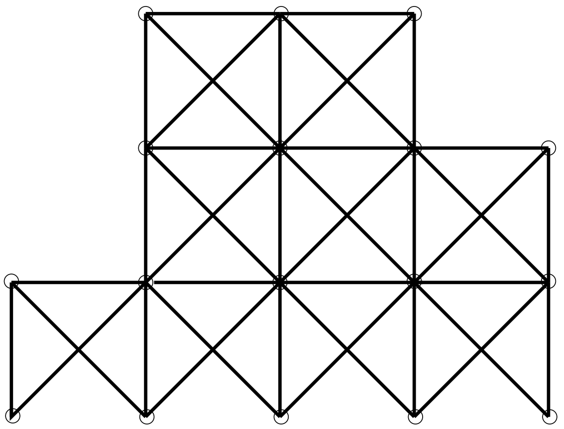


**Figure 2:** Truss structure.

we minimize the $\ell_1$ 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 search method for computing the optimum sparsity is impractical). The resulting LP that must be solved to find the

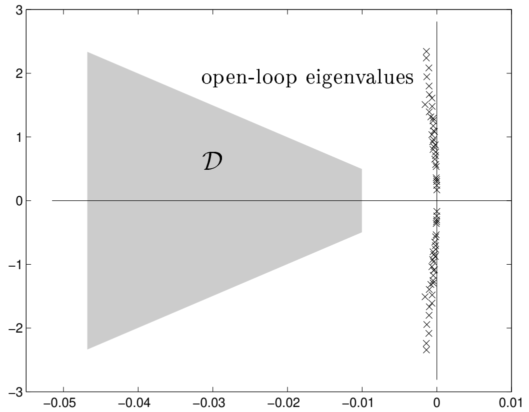



**Figure 3:** Desired region $\mathcal{D}$ for closed-loop eigenvalues.

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 using the LP solver `PCx`[1] on a typical personal computer). The locations of the nonzero dampers are shown in Figure 4 (a solid line between two nodes 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, and we get $\sum_i b_i = 1.73$ and $\max_i b_i = 0.27$ which are a measure of the total amount of damping material that must be added to the structure and to a single strut.

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

[1] `http://www-c.mcs.anl.gov/home/otc/Library/PCx/`

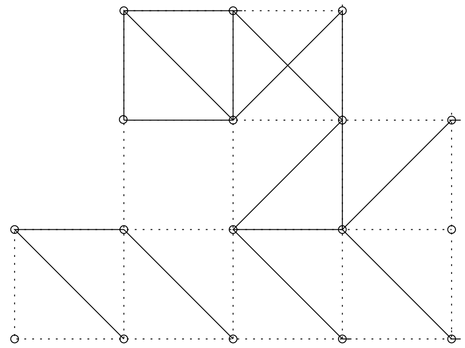


**Figure 4:** Locations of dampers.

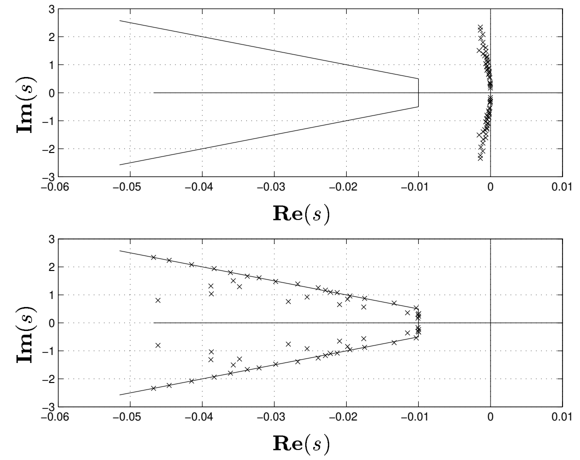



**Figure 5:** Open-loop and closed-loop eigenvalues.

### 7.2 LAC using 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 horizontal and vertical rate sensor can be placed at each node 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 via a feedback gain that must be determined. The dynamics of the closed-loop system are written as $\dot{z} = A(x)z$ such that the vector of design variables $x \in \mathbf{R}^{1326}$ represents the elements of the $34 \times 39$ matrix of feedback gains from each sensor to each actuator. In this case also, $A(x)$ is affine in $x$.

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 $\ell_1$ norm of $x$ subject to the eigenvalue placement specifications, which is an LP with 1326 variables and 64 linear inequality constraints (this takes approximately a minute using `PCx` on a typical personal computer).

The sparsity pattern of the resulting feedback gain matrix is given in Figure 6. Again, the solution is very sparse and only 16 out of 1326 possible feedback gains are nonzero ($\sum_i |x_i| = 3.44$ and $\max_i |x_i| = 0.56$). Of course, actuators (sensors) that are not connected to a sensor (actuator) can be eliminated. For the solution

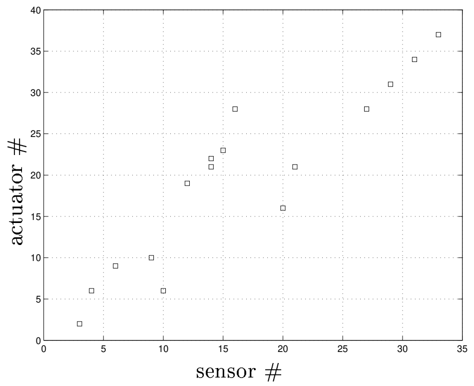



**Figure 6:** Sparsity pattern of the feedback gain matrix. A non-zero feedback gain from that sensor to actuator is shown by a '□'.

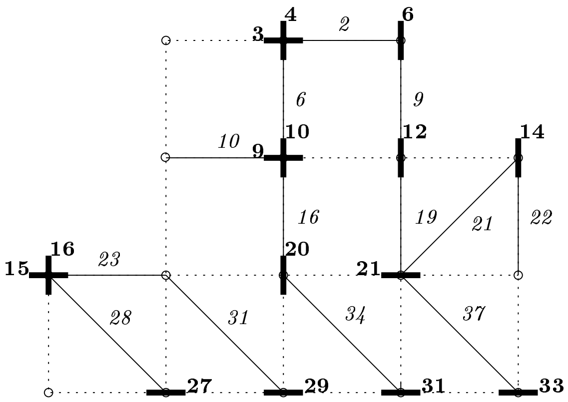



**Figure 7:** Sensor & actuator locations. A solid line between two nodes corresponds to an actuator between those two nodes. A vertical or horizontal line crossing a node corresponds to a vertical or horizontal rate sensor at that node. The actuator numbers are *italicized* and the sensor numbers are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

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 Figures 6 and 7 it can be seen that the controller is colocated 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, while 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. Computing the closed-loop eigenvalues for this design show that the closed-loop eigenvalues meet or exceed the design specifications, which verifies the low-authority assumption in this design approach.

As a final remark, it should be noted that the examples given here are the simplest of the type. More sophisticated ones (including robust LAC, and combined dynamic HAC/LAC) are considered in [12]. Furthermore, the solution time scales very well with the problem size.

## 8 Conclusions

In this paper we addressed the problem of robust and sparse LAC design using convex optimization. The main points were:

- LP, SOCP, and SDP can be used to solve very complex LAC problems, involving complicated cases with substantial spillover. LP, SOCP, and SDPs can be solved (globally) for huge problem sizes.
- In *many* applications it is desirable to compute a sparse $x$ which can be done by an $\ell_1$ relaxation method. These applications include actuator/sensor placement and controller topology design.

Different LAC design constraints in previous sections can be mixed freely. These constraints were either linear inequalities, second-order cone constraints, or LMIs in $x$, and therefore, an SDP solver (*e.g.*, [9]) can be used to compute the desired $x$. In other words, multiobjective LAC design can be handled efficiently in practice. By minimizing $\|x\|_1$ subject to these design specifications, it is possible, for example, to (heuristically) solve the actuator/sensor placement problem as well.

### References


[1] J. H. Wykes. Structural dynamic stability augmentation and gust alleviation of flexible aircraft. In *AIAA Paper 68-1067, AIAA Anual Meeting*, October 1968.

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

[3] 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.

[4] Y. Zhang. *User's guide to LIPSOL: a matlab toolkit for linear programming interior-point solvers.* Math. & Stat. Dept., Univ. Md. Baltimore County, October 1994. Beta release.

[5] J. Czyzyk, S. Mehrotra, and S. J. Wright. *PCx User Guide.* Optimization Technology Cneter, March 1997.

[6] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and Appl.*, 1997. To appear.

[7] M. S. Lobo, L. Vandenberghe, and S. Boyd. SOCP*: Software for Second-Order Cone Programming.* Information Systems Laboratory, Stanford University, 1997.

[8] T. Kato. *A Short Introduction to Perturbation Theory for Linear Operators.* Springer-Verlag, 1982.

[9] 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.

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

[11] F. Alizadeh, J. P. Haeberly, M. V. Nayakkankuppam, M. L. Overton, and S. Schmieta. SDPPACK *User's Guide, Version 0.9 Beta.* NYU, June 1997.

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

[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.