43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Robust Stabilization of Discrete-Time Systems

Jesse B. Hoagg[1] and Dennis S. Bernstein[2]

Abstract— **This paper considers the robust stabilization of single-input single-output (SISO) linear shift-invariant discrete-time systems. A non-minimal state-space realization is used to transform the original SISO system into a higher order realization with full-state measurement without knowledge of the system parameters. Robust stabilization is achieved using a weighted least squares optimization with the stability constraint implemented as linear matrix inequalities.**

1. Introduction

Robust stability analysis has been widely studied in various settings using diverse techniques. For polynomials, Kharitonov's Theorem shows that interval polynomials are Hurwitz if and only if four special vertex polynomials are Hurwitz [1, 2]. The Edge Theorem provides necessary and sufficient conditions for the stability of a polytope of polynomials [3, 4]. Stability analysis of interval matrices remains a problem of interest [5, 6]. Although Kharitonov's Theorem does not hold for discrete-time systems [7], the delta-operator has been used to develop Kharitonov-like theorems for discrete-time stability [8, 9].

For robust stabilization, linear matrix inequalities and convex optimization methods are used in [10] to simultaneously stabilize a finite number of discrete-time systems. In the present paper, we use vertex linear matrix inequalities and convex optimization to robustly stabilize a convex set of uncertain discrete-time systems.

We consider the robust stabilization of single-input single-output (SISO) linear shift-invariant discrete-time systems. We recast the output feedback problem as a full-state-feedback problem using the non-minimal state-space representation given in [11]. Vertex matrix inequalities are then used to enforce the stability of the uncertain closed-loop system. For controller synthesis, robust stabilization is achieved by means of a weighted least squares convex optimization problem with the stability constraint implemented as linear matrix inequalities.

In Section 2, robust dynamic compensation is formulated as a optimization problem. A non-minimal state-space representation is given in Section 3. In Section 4, the robust dynamic compensation problem is recast as a robust full-state feedback problem using the non-minimal state-space representation. Algorithm implementation is discussed in Section 5. Numerical examples are presented in Section 6. Conclusions are given in Section 7.

[1]National Defense Science and Engineering Graduate Fellow, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109-2140, phone: (734) 763-1305, fax: (734) 763-0578, email: jhoagg@umich.edu

[2]Professor, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI 48109-2140, email: dsbaero@umich.edu

2. Robust Dynamic Compensation

We consider the discrete-time single-input single-output linear shift-invariant system

$$y_k = \frac{b(z)}{a(z)} u_k, \tag{2.1}$$

where

$$b(z) \triangleq b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \cdots + b_1 z + b_0, \tag{2.2}$$

$$a(z) \triangleq z^n + a_{n-1}z^{n-1} + a_{n-2}z^{n-2} + \cdots + a_1 z + a_0, \tag{2.3}$$

and b_i and a_i are constant for $i = 0, \ldots, n-1$. Note that (2.1)-(2.3) includes all strictly proper linear discrete-time systems with relative degree ρ by letting $\rho - 1$ leading coefficients of $b(z)$ equal zero. We make the following assumptions.

(i) The polynomials $a(z)$ and $b(z)$ are coprime.
(ii) The system order n is known.
(iii) For all $i = 0, \ldots, n-1$, there exist known $\hat{a}_i$, $\hat{b}_i$, α_i, β_i such that

$$a_i = \hat{a}_i + \gamma_i, \qquad b_i = \hat{b}_i + \delta_i, \tag{2.4}$$

where $\gamma_i \in [-\alpha_i, +\alpha_i]$ and $\delta_i \in [-\beta_i, +\beta_i]$.

The objective of this paper is to design a robust dynamic compensator that stabilizes a large class of systems characterized by (2.1)-(2.3) under assumptions *(i)*-*(iii)*. We therefore formulate robust stabilization as an optimization problem.

The system (2.1)-(2.3) has the state-space realization

$$x_{k+1} = (A + B\Delta_A)x_k + Bu_k, \tag{2.5}$$

$$y_k = (C + \Delta_C)x_k, \tag{2.6}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times 1}$, $C \in \mathbb{R}^{1\times n}$, $\Delta_A \in \mathbb{R}^{1\times n}$, and $\Delta_C \in \mathbb{R}^{1\times n}$ are given by

$$A \triangleq \begin{bmatrix} -\hat{a}_{n-1} & \cdots & -\hat{a}_1 & -\hat{a}_0 \\ 1 & & 0 & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & 1 & 0 \end{bmatrix}, \quad B \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.7}$$

$$C \triangleq \begin{bmatrix} \hat{b}_{n-1} & \cdots & \hat{b}_1 & \hat{b}_0 \end{bmatrix}, \tag{2.8}$$

$$\Delta_A \triangleq \sum_{i=0}^{n-1} -\gamma_i e_i, \qquad \Delta_C \triangleq \sum_{i=0}^{n-1} \delta_i e_i, \tag{2.9}$$

where, for $i = 0, \ldots, n-1$, $e_i \triangleq \begin{bmatrix} 0_{1\times(n-1-i)} & 1 & 0_{1\times i} \end{bmatrix}$. Furthermore, we consider the dynamic compensator

$$z_{k+1} = A_c z_k + B_c y_k, \tag{2.10}$$
$$u_k = C_c z_k, \tag{2.11}$$

where $A_c \in \mathbb{R}^{n\times n}$, $B_c \in \mathbb{R}^{n\times 1}$, and $C_c \in \mathbb{R}^{1\times n}$. Without loss of generality, we express the dynamic compensator in the controllable canonical form

$$A_c \triangleq \begin{bmatrix} & a_c & \\ I_{n-1} & & 0_{(n-1)\times 1} \end{bmatrix}, \quad B_c \triangleq \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.12}$$

where $a_c \in \mathbb{R}^{1\times n}$. The closed-loop system

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A + B\Delta_A & BC_c \\ B_c\left(C + \Delta_C\right) & A_c \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} \tag{2.13}$$

is asymptotically stable if and only if there exists $P > 0$ such that

$$P - \left(\begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix} + \begin{bmatrix} B\Delta_A & 0 \\ B_c\Delta_C & 0 \end{bmatrix}\right) P \times \left(\begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix} + \begin{bmatrix} B\Delta_A & 0 \\ B_c\Delta_C & 0 \end{bmatrix}\right)^{\mathrm{T}} > 0. \tag{2.14}$$

The following result concerns vertex matrix inequalities.

Lemma 2.1. *Let ν and r be positive integers, and let $\mathcal{A} \in \mathbb{R}^{\nu\times\nu}$. For $i = 1, \ldots, r$, let $\eta_i > 0$ and let $\mathcal{A}_i \in \mathbb{R}^{\nu\times\nu}$. Define the uncertainty set*

$$\mathcal{U} \triangleq \left\{\Delta\mathcal{A} : \Delta\mathcal{A} = \sum_{i=1}^{r} \eta_i \mathcal{A}_i, \text{ where } \eta_i \in [-\sigma_i, +\sigma_i], i = 1, \ldots, r\right\} \tag{2.15}$$

and let $P > 0$. Then P satisfies

$$P - (\mathcal{A} + \Delta\mathcal{A})\, P\, (\mathcal{A} + \Delta\mathcal{A})^{\mathrm{T}} > 0, \tag{2.16}$$

for all $\Delta A \in \mathcal{U}$ if and only if P satisfies the 2^r matrix inequalities

$$P - (\mathcal{A} \pm \sigma_1\mathcal{A}_1 \pm \cdots \pm \sigma_r\mathcal{A}_r)\, P \times (\mathcal{A} \pm \sigma_1\mathcal{A}_1 \pm \cdots \pm \sigma_r\mathcal{A}_r)^{\mathrm{T}} > 0. \tag{2.17}$$

Proof. Let $\Delta\mathcal{A}_1, \ldots, \Delta\mathcal{A}_{2^r}$ be the 2^r matrices $\pm\gamma_1\mathcal{A}_1 \pm \cdots \pm \gamma_r\mathcal{A}_r$. For $i = 1, \ldots, 2^r$, assume that $P > 0$ satisfies the matrix inequalities

$$P - (\mathcal{A} + \Delta\mathcal{A}_i)\, P\, (\mathcal{A} + \Delta\mathcal{A}_i)^{\mathrm{T}} > 0. \tag{2.18}$$

Using Schur complements, (2.18) is equivalent to

$$\begin{bmatrix} P & P\left(\mathcal{A} + \Delta\mathcal{A}_i\right)^{\mathrm{T}} \\ \left(\mathcal{A} + \Delta\mathcal{A}_i\right) P & P \end{bmatrix} > 0. \tag{2.19}$$

The inequalities (2.19) are linear in $\Delta\mathcal{A}_i$ and $\mathcal{U}$ is the convex hull of $\Delta\mathcal{A}_1, \ldots, \Delta\mathcal{A}_r$. Thus, we conclude that

$$P - (\mathcal{A} + \Delta\mathcal{A})\, P\, (\mathcal{A} + \Delta\mathcal{A})^{\mathrm{T}} > 0, \tag{2.20}$$

for all $\Delta\mathcal{A} \in \mathcal{U}$. The converse result follows immediately from the fact that $\Delta\mathcal{A}_i \in \mathcal{U}$ for all $i = 1, \ldots, 2^n$. □

In Lemma 2.1, the vertex matrix inequalities characterize a single Lyapunov function $V(x) = x^{\mathrm{T}}Px$ for all uncertainty in the set $\mathcal{U}$. Hence the Lyapunov function guarantees stability for time-varying uncertainty. To avoid cumbersome notation, we consider only time-invariant uncertainty, although the method applies to the time-varying case.

Lemma 2.1 implies that the inequality (2.14) is equivalent to the 2^{2n} matrix inequalities

$$P - \left(\begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix} + \begin{bmatrix} B\Delta_{\alpha i} & 0 \\ B_c\Delta_{\beta j} & 0 \end{bmatrix}\right) P \times \left(\begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix} + \begin{bmatrix} B\Delta_{\alpha i} & 0 \\ B_c\Delta_{\beta j} & 0 \end{bmatrix}\right)^{\mathrm{T}} > 0, \tag{2.21}$$

where $\Delta_{\alpha 1}, \ldots, \Delta_{\alpha 2^n}$ denote the 2^n row vectors $\pm\alpha_0 e_0 \pm \cdots \pm \alpha_{n-1}e_{n-1}$, and $\Delta_{\beta 1}, \ldots, \Delta_{\beta 2^n}$ denote the 2^n row vectors $\pm\beta_0 e_0 \pm \cdots \pm \beta_{n-1}e_{n-1}$.

Now we formulate the robust stabilization of (2.5)-(2.11) as an optimization problem. We define the optimization variables $\hat{\Delta}_{\alpha 1}, \ldots, \hat{\Delta}_{\alpha 2^n} \in \mathbb{R}^{1\times n}$ and $\hat{\Delta}_{\beta 1}, \ldots, \hat{\Delta}_{\beta 2^n} \in \mathbb{R}^{1\times n}$. Additional optimization variables include a_c, C_c, and P. The objective is to minimize

$$J(\hat{\Delta}_{\alpha 1}, \ldots, \hat{\Delta}_{\alpha 2^n}, \hat{\Delta}_{\beta 1}, \ldots, \hat{\Delta}_{\beta 2^n}) = \sum_{i=1}^{2^n}\sum_{j=1}^{2^n} \left\| W_1 \left(\begin{bmatrix} \hat{\Delta}_{\alpha i}^{\mathrm{T}} \\ \hat{\Delta}_{\beta j}^{\mathrm{T}} \end{bmatrix} - \begin{bmatrix} \Delta_{\alpha i}^{\mathrm{T}} \\ \Delta_{\beta j}^{\mathrm{T}} \end{bmatrix} \right) \right\|_F^2, \tag{2.22}$$

subject to the 2^{2n} stability constraints

$$P - \Phi_{ij} P \Phi_{ij}^{\mathrm{T}} > 0, \tag{2.23}$$

where for $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$,

$$\Phi_{ij} \triangleq \begin{bmatrix} A & BC_c \\ B_cC & A_c \end{bmatrix} + \begin{bmatrix} B\hat{\Delta}_{\alpha i} & 0 \\ B_c\hat{\Delta}_{\beta j} & 0 \end{bmatrix}, \tag{2.24}$$

and W_1 is a weighting matrix. This constrained least-squares problem seeks to minimize the distance between the achievable bounds $\hat{\Delta}_{\alpha i}$ and $\hat{\Delta}_{\beta j}$ and the desired bounds $\Delta_{\alpha i}$ and $\Delta_{\beta j}$ for $i, j = 1, \ldots, 2^n$ subject to the stability constraints (2.23).

Using Schur complements, (2.23) is equivalent to the matrix inequality

$$\begin{bmatrix} P & \Phi_{ij}P \\ P\Phi_{ij}^{\mathrm{T}} & P \end{bmatrix} > 0. \tag{2.25}$$

The terms $\begin{bmatrix} 0 & BC_c \\ 0 & A_c \end{bmatrix} P$ and $\begin{bmatrix} B\hat{\Delta}_{\alpha i} & 0 \\ B_c\hat{\Delta}_{\beta j} & 0 \end{bmatrix} P$ in (2.25) are quadratic in the optimization variables. One approach to dealing with these terms is to define the change of variables $S \triangleq \begin{bmatrix} 0 & BC_c \\ 0 & A_c \end{bmatrix} P$ and $T \triangleq \begin{bmatrix} B\hat{\Delta}_{\alpha i} & 0 \\ B_c\hat{\Delta}_{\beta j} & 0 \end{bmatrix} P$, and optimize with respect to S, T, and P. However, since S and T require structural constraints, the optimization will not necessarily yield S, T, and P that can be solved for the variables of interest $\hat{\Delta}_{\alpha i}$, $\hat{\Delta}_{\beta j}$, A_c, and C_c. Thus, we consider an alternative method for implementing the stability inequalities as convex constraints.

In the next section, we reformulate the output feedback problem as a full-state feedback problem using a non-minimal state-space representation that provides full-state measurement even with parameter uncertainty. Then we implement stability inequalities using full-state feedback as a convex constraint for use in a weighted least-squares optimization.

3. Obtaining Full-State Measurement

In this section, we consider a non-minimal state-space realization of (2.1)-(2.3) where the state-vector contains only delayed values of the input and output [11]. Furthermore, we achieve state estimation in the non-minimal basis using a deadbeat observer, which does not require knowledge of the parameters $a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}$. The system (2.1)-(2.3) has the non-minimal state-space realization

$$x_{k+1} = Ax_k + Bu_k, \tag{3.1}$$

$$y_k = Cx_k, \tag{3.2}$$

where $A \in \mathbb{R}^{2n\times 2n}$, $B \in \mathbb{R}^{2n\times 1}$, and $C \in \mathbb{R}^{1\times 2n}$ are given by

$$A \triangleq \begin{bmatrix} -a_{n-1} & \cdots & -a_1 & -a_0 & b_{n-1} & \cdots & b_1 & b_0 \\ 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & & & & \vdots \\ 0 & & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & & & 0 \\ \vdots & & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & & 1 & 0 \end{bmatrix}, \tag{3.3}$$

$$B \triangleq \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}, \tag{3.4}$$

$$C \triangleq \begin{bmatrix} -a_{n-1} & \cdots & -a_0 & b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix}, \tag{3.5}$$

$$x_k \triangleq \begin{bmatrix} y_{k-1} & \cdots & y_{k-n} & u_{k-1} & u_{k-2} & \cdots & u_{k-n} \end{bmatrix}^{\mathrm{T}}. \tag{3.6}$$

The state-vector x_k contains only delayed values of the input and output. Thus full-state measurement is obtained by recording n previous input and output data points. Recording the n previous data points is equivalent to implementing the state estimator

$$\hat{x}_{k+1} = A_e\hat{x}_k + Bu_k + F_e y_k, \tag{3.7}$$

where

$$F_e \triangleq \begin{bmatrix} 1 \\ 0_{(2n-1)\times 1} \end{bmatrix}, \qquad A_e \triangleq A - F_e C \tag{3.8}$$

Note that A_e is nilpotent. In particular, $A_e^n = 0$. Thus, (3.7) is deadbeat after n time steps, and $\hat{x}_k = x_k$ for all $k \geq n$.

Using the non-minimal state-space representation given by (3.1)-(3.6), we recast an n-dimensional system with output measurement as a $2n$-dimensional system with full-state measurement. Now we examine the controllability and observability properties of the realization (3.1)-(3.6) with full-state measurement. Clearly, the pair (A, I) is completely observable.

Proposition 3.1. *Consider the state-space realization (3.1)-(3.6). The triple (A, B, I) is completely controllable if and only if the polynomials $a(z)$ and $b(z)$ are coprime.*

Proof. Define

$$\mathcal{C}(z) \triangleq \begin{bmatrix} zI - A & B \end{bmatrix}. \tag{3.9}$$

Assume (A, B) is controllable, which is equivalent to

$$\operatorname{rank} \mathcal{C}(z) = 2n, \tag{3.10}$$

for all $z \in \mathbb{C}$. Let

$$\mathcal{C}(z) = \begin{bmatrix} \mathcal{C}_1(z) & \mathcal{C}_2(z) \end{bmatrix}, \tag{3.11}$$

where $\mathcal{C}_1(z) \in \mathbb{C}^{2n\times n}$ and $\mathcal{C}_2(z) \in \mathbb{C}^{2n\times(n+1)}$, so that (3.10) can be written as

$$\operatorname{rank} \begin{bmatrix} \mathcal{C}_1(z) & \mathcal{C}_2(z) \end{bmatrix} = 2n, \tag{3.12}$$

for all $z \in \mathbb{C}$. Note that the columns of $\mathcal{C}_1(z)$ are linearly independent of the columns of $\mathcal{C}_2(z)$ for all z. Furthermore, note that rank $\mathcal{C}_1(z) \geq n-1$ and rank $\mathcal{C}_2(z) \geq n$. Thus (3.12) is equivalent to either

$$\operatorname{rank} \mathcal{C}_1(z) = n \tag{3.13}$$

or

$$\operatorname{rank} \mathcal{C}_2(z) = n + 1, \tag{3.14}$$

for all z. Now we examine the two conditions (3.13) and (3.14) separately. Note that rank $\mathcal{C}_1(z) = n$ for all z that is not a root of $a(z)$, while rank $\mathcal{C}_2(z) = n+1$ for all z that is not a root of $b(z)$. Therefore, (3.12) holds if and only if $a(z)$ and $b(z)$ are coprime. The converse result follows from reversing these steps. □

For the remainder of this paper, we assume the use of the deadbeat observer (3.7) to record n previous input and output data points for use in feedback.

4. Robust Stabilization with Full-State Feedback

In this section, we consider the robust stabilization problem of Section 1. We transform the problem to robust full-state feedback stabilization by using the non-minimal state-space representation (3.1)-(3.6). Define

$$\hat{A} \triangleq \begin{bmatrix} -\hat{a}_{n-1} & \cdots & -\hat{a}_1 & -\hat{a}_0 & \hat{b}_{n-1} & \cdots & \hat{b}_1 & \hat{b}_0 \\ 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & & & & \vdots \\ 0 & & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \cdots & 0 & 0 & 1 & & & 0 \\ \vdots & & & \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 0 & 0 & & 1 & 0 \end{bmatrix}, \tag{4.1}$$

$$D \triangleq \begin{bmatrix} 1 & 0_{(2n-1)\times 1} \end{bmatrix}^{\mathrm{T}}, \tag{4.2}$$

so that (3.1) can be written as

$$x_{k+1} = \left(\hat{A} + D \begin{bmatrix} \Delta_A & \Delta_C \end{bmatrix}\right) x_k + B u_k. \tag{4.3}$$

Consider the full-state feedback controller

$$u_k = K x_k, \tag{4.4}$$

where $K \in \mathbb{R}^{1\times 2n}$, so that the closed-loop system is given by

$$x_{k+1} = \left(\hat{A} + D \begin{bmatrix} \Delta_A & \Delta_C \end{bmatrix} + BK\right) x_k. \tag{4.5}$$

The closed-loop system (4.5) is asymptotically stable if and only if there exists $P > 0$ such that

$$P - \left(\hat{A} + D \begin{bmatrix} \Delta_A & \Delta_C \end{bmatrix} + BK\right) P \times \left(\hat{A} + D \begin{bmatrix} \Delta_A & \Delta_C \end{bmatrix} + BK\right)^{\mathrm{T}} > 0. \tag{4.6}$$

Lemma 2.1 implies that (4.6) is equivalent to the 2^{2n} matrix inequalities

$$P - \left(\hat{A} + D \begin{bmatrix} \Delta_{\alpha i} & \Delta_{\beta j} \end{bmatrix} + BK\right) P \times \left(\hat{A} + D \begin{bmatrix} \Delta_{\alpha i} & \Delta_{\beta j} \end{bmatrix} + BK\right)^{\mathrm{T}} > 0, \tag{4.7}$$

for $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$. The optimization variables are $\hat{\Delta}_{\alpha 1}, \ldots, \hat{\Delta}_{\alpha 2^n}$, $\hat{\Delta}_{\beta 1}, \ldots, \hat{\Delta}_{\beta 2^n}$, P, and K. The objective is to minimize (2.22) subject to the 2^{2n} stability constraints

$$P - \left(\hat{A} + D \begin{bmatrix} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{bmatrix} + BK\right) P \times \left(\hat{A} + D \begin{bmatrix} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{bmatrix} + BK\right)^{\mathrm{T}} > 0, \tag{4.8}$$

where $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$.

We now express the stability constraints (4.8) as linear matrix inequalities for use in convex optimization. Using Schur complements, (4.8) is equivalent to

$$\begin{bmatrix} P & \Omega_{ij} P \\ P \Omega_{ij}^{\mathrm{T}} & P \end{bmatrix} > 0, \tag{4.9}$$

where

$$\Omega_{ij} \triangleq \hat{A} + D \begin{bmatrix} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{bmatrix} + BK. \tag{4.10}$$

Since (4.9) involves the quadratic terms KP and $\begin{bmatrix} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{bmatrix} P$, we define

$$R \triangleq KP, \qquad S_{ij} \triangleq \begin{bmatrix} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{bmatrix} P, \tag{4.11}$$

for $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$, so that (4.9) can be written as

$$\begin{bmatrix} P & \hat{A}P + DS_{ij} + BR \\ P\hat{A}^{\mathrm{T}} + S_{ij}^{\mathrm{T}} D^{\mathrm{T}} + R^{\mathrm{T}} B^{\mathrm{T}} & P \end{bmatrix} > 0, \tag{4.12}$$

where $P > 0$, $i = 1, \ldots, 2^n$, and $j = 1, \ldots, 2^n$. The 2^{2n} linear matrix inequalities given by conditions (4.11)-(4.12) are equivalent to the stability conditions (4.8).

To impose the discrete-time stability constraints (4.11)-(4.12) on the cost function (2.22) we define $W_1 \triangleq P$ so that (2.22) becomes

$$J(S_{11}, \ldots, S_{2^n 2^n}, P) = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \left\| S_{ij}^{\mathrm{T}} - P \begin{bmatrix} \Delta_{\alpha i}^{\mathrm{T}} \\ \Delta_{\beta j}^{\mathrm{T}} \end{bmatrix} \right\|_F^2. \tag{4.13}$$

Equation (4.13) and the 2^{2n} discrete-time stability constraints (4.11)-(4.12) constitute a constrained least-squares optimization, which is linear in the optimization parameters P, R, and S_{ij} for $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$. We relax the constraints (4.12) to

$$\begin{bmatrix} P & \hat{A}P + DS_{ij} + BR \\ P\hat{A}^{\mathrm{T}} + S_{ij}^{\mathrm{T}} D^{\mathrm{T}} + R^{\mathrm{T}} B^{\mathrm{T}} & P \end{bmatrix} \geq \sigma I, \tag{4.14}$$

where $\sigma > 0$ is arbitrarily small so that the optimization is convex.

The cost function (4.13) does not require that the achievable bounds $\hat{\Delta}_{\alpha i}$ and $\hat{\Delta}_{\beta j}$ be within the the polytope defined by the desired bounds $\Delta_{\alpha i}$ and $\Delta_{\beta j}$. Hence, if the desired bounds cannot be achieved, the resulting bounds, although minimal in the sense to the cost function (4.13), may define a polytope of little interest for the robust stabilization problem. Thus, we adopt an iterative approach to finding the maximum achievable bounds for robust stabilization.

5. Algorithm Implementation

The constrained least squares optimization problem is to minimize

$$J(S_{11}, \ldots, S_{2^n 2^n}, P) = \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \left\| S_{ij}^{\mathrm{T}} - P \begin{bmatrix} \Delta_{\alpha i}^{\mathrm{T}} \\ \Delta_{\beta j}^{\mathrm{T}} \end{bmatrix} \right\|_F^2, \tag{5.1}$$

subject to

$$\begin{bmatrix} P & \hat{A}P + DS_{ij} + BR \\ P\hat{A}^{\mathrm{T}} + S_{ij}^{\mathrm{T}} D^{\mathrm{T}} + R^{\mathrm{T}} B^{\mathrm{T}} & P \end{bmatrix} \geq \sigma I, \tag{5.2}$$

where

$$P = P^{\mathrm{T}} > 0 \tag{5.3}$$

and $\sigma > 0$ is arbitrarily small.

The resulting vertices are determined by

$$\left[\begin{array}{cc} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{array} \right] = S_{ij} P^{-1}, \tag{5.4}$$

where $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$. The stability constraints (5.2) require that the closed-loop system is asymptotically stable for all uncertainty contained in the convex hull of (5.4).

Consider the desired vertices $\Delta_{\alpha i}$ and $\Delta_{\beta j}$, and let $\epsilon \in [0, 1]$. We perform the above optimization with some arbitrarily small ϵ such that

$$\hat{\Delta}_{\alpha i} = \epsilon \Delta_{\alpha i}, \tag{5.5}$$

$$\hat{\Delta}_{\beta j} = \epsilon \Delta_{\beta j}, \tag{5.6}$$

for $i = 1, \ldots, 2^n$ and $j = 1, \ldots, 2^n$. Next, the optimization is iteratively performed while ϵ is increased by small increments. The maximal bounds are determined by the largest value of ϵ for which the equations (5.5)-(5.6) hold. The stabilizing gain is determined by

$$K = RP^{-1}. \tag{5.7}$$

The optimization is performed using the SeDuMi Matlab toolbox [12]. SeDuMi solves linear programming problems over symmetric cones, allowing us to impose quadratic and positive semi-definite constraints.

6. Examples

6.1. The scalar case

Consider the scalar discrete-time system

$$y_k = \frac{b_0 + \delta_0}{z + (a_0 + \gamma_0)} u_k, \tag{6.1}$$

where $a_0 = 2.0$, $b_0 = 3.0$, $\gamma_0 \in [-0.4, 0.4]$, and $\delta_0 \in [-0.6, 0.6]$. The desired vertices $\left[\begin{array}{cc} \Delta_{\alpha i} & \Delta_{\beta j} \end{array} \right]$ for $i = 1, 2$ and $j = 1, 2$ are given by the 4 row vectors $\left[\begin{array}{cc} \pm 0.4 & \pm 0.6 \end{array} \right]$.

The iterative optimization presented given in Section 5 is performed on the uncertain system (6.1). The vertex points obtained by the optimization are

$$\left[\begin{array}{cc} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{array} \right] = \left[\begin{array}{cc} \pm 0.22 & \pm 0.33 \end{array} \right]. \tag{6.2}$$

Figure 1 show the desired stability region and the guaranteed stability region achieved by the optimization.

Therefore, the feedback $u_k = K x_k$ stabilizes (6.1) for all $\gamma_0 \in [-0.22, 0.22]$ and $\delta_0 \in [-0.33, 0.33]$. The gain matrix obtain from the optimization is

$$K = \left[\begin{array}{cc} -1.387 & 2.079 \end{array} \right]. \tag{6.3}$$

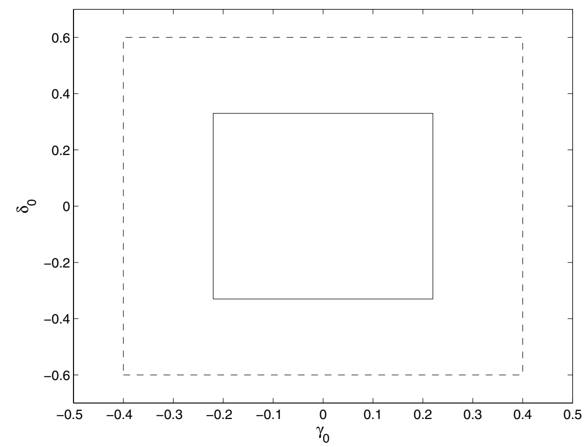

Fig. 1. The desired stability region (dashed) and the achieved stability region (solid).

6.2. Discrete-time harmonic oscillator

Consider the discrete-time simple harmonic oscillator

$$y_k = \frac{b_1 z + b_0}{z^2 - 2z + (\omega^2 T_s^2 + 1)} u_k, \tag{6.4}$$

where b_1, b_0, and ω are uncertain and $T_s = 0.01$. The uncertain parameters are

$$\omega = 45 + \delta_\omega, \tag{6.5}$$

$$b_0 = 4.0 + \delta_0, \tag{6.6}$$

$$b_1 = 2.0 + \delta_1, \tag{6.7}$$

where $\delta_\omega \in [-13, 10]$, $\delta_0 \in [-2, 2]$, $\delta_1 \in [-1, 1]$. The discrete-time system (6.4) has the non-minimal state-space realization

$$x_{k+1} = \left[\begin{array}{cccc} 2 & -1.2025 - \gamma_0 & 2.0 + \delta_1 & 4.0 + \delta_0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right] x_k + \left[\begin{array}{cccc} 0 & 0 & 1 & 0 \end{array} \right]^{\mathrm{T}} u_k, \tag{6.8}$$

$$y_k = \left[\begin{array}{cccc} 2 & -1.2025 - \gamma_0 & 2.0 + \delta_1 & 4.0 + \delta_0 \end{array} \right], \tag{6.9}$$

where $\gamma_0 \in [-0.1, 0.1]$. Therefore, the vertices $\left[\begin{array}{cc} \Delta_{\alpha i} & \Delta_{\beta j} \end{array} \right]$ for $i = 1, \ldots, 4$ and $j = 1, \ldots, 4$ are given by the 16 row vectors $\left[\begin{array}{cccc} \pm 0 & \pm 0.1 & \pm 1 & \pm 2 \end{array} \right]$.

The iterative optimization is performed on the uncertain system (6.8)-(6.9). The vertex points obtained by the optimization are

$$\left[\begin{array}{cc} \hat{\Delta}_{\alpha i} & \hat{\Delta}_{\beta j} \end{array} \right] = \left[\begin{array}{cccc} \pm 0 & \pm 0.07 & \pm 0.7 & \pm 1.4 \end{array} \right]. \tag{6.10}$$

Figure 2 show the desired stability region and the guaranteed stability region achieved by the optimization.

Thus, the feedback $u_k = K x_k$ stabilizes (6.8)-(6.9) for all $\delta_\omega \in [-8.6, 7.2]$, $\delta_0 \in [-0.7, 0.7]$, and $\delta_1 \in [-1.4, 1.4]$. The gain matrix obtain from the optimization is

$$K = \left[\begin{array}{cccc} -0.2580 & 0.3042 & -1.338 & -1.012 \end{array} \right]. \tag{6.11}$$

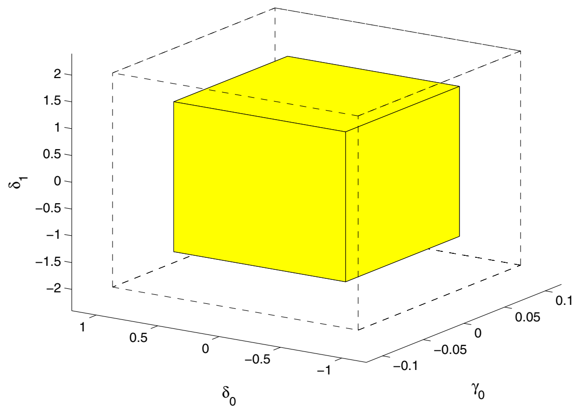

Fig. 2. The desired stability region (dashed) and the achieved stability region (shaded).

7. Conclusions

In this paper, we presented a method for robustly stabilizing single-input single-output discrete-time systems. The method relies on a non-minimal state-space realization and vertex matrix inequalities. Lyapunov stability conditions were implemented as linear matrix inequalities for a weighted least squares optimization.

References

[1] V. Kharitonov, "Asymptotic stability of an equilibrium position of a family of systems of linear differential equations," *Differential'nye Uravnenlya*, vol. 14, no. 11, pp. 1483–1485, 1978.

[2] I. Petersen, "A new extension to Kharitonov's theorem," *IEEE Trans. Autom. Contr.*, vol. 35, no. 7, pp. 825–828, 1990.

[3] A. Bartlett, C. Hollot, and H. Lin, "Root locations of an entire polytope of polynomials: It suffices to check the edges," in *Proc. Amer. Cont. Conf.*, Minneapolis, 1987.

[4] M. Fu and B. Barmish, "Polytope of polynomials with zeros in a prescribed set," *IEEE Trans. Autom. Contr.*, vol. 34, pp. 544–546, 1989.

[5] R. Yedavalli, "Conditions for the exstence of a common quadratic Lyapunov function via stability analysis of matrix families," in *Proc. Amer. Cont. Conf.*, 2002, pp. 1296–1301.

[6] ——, "It suffices to check only two special vertex matrices in Kronecker space to analyze the robust stability of an interval matrix," in *Proc. Amer. Cont. Conf.*, 2002, pp. 1266–1271.

[7] C. Hollot and A. Bartlett, "Some discrete-time counterparts to Kharitonov's stability criterion for uncertain systems," *IEEE Trans. Autom. Contr.*, vol. 31, no. 4, pp. 355–356, 1986.

[8] A. Feuer and E. Zeheb, "Robust stability of interval polynomials in discrete-time," in *Proc. IEEE Conf. on Dec. and Cont.*, San Diego, CA, Dec. 1997, pp. 902–907.

[9] I. Petersen, "A class of stability regions for which a Kharitonov-like theorem holds," *IEEE Trans. Autom. Contr.*, vol. 34, no. 10, pp. 1111–1115, 1989.

[10] J. Kamali and G. Franklin, "Simultaneous controller design: A linear matrix inequalities approach," in *Proc. IEEE Inter. Conf. on Syst., Man, and Cybernetics*, Vancouver, BC Canada, Oct. 1995, pp. 553–556.

[11] J. Daams and J. Polderman, "Almost optimal adaptive lq control: SISO case," *Mathematics of Control, Signals, and Systems*, vol. 15, no. 15, pp. 71–100, 2002.

[12] J. Strum, "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones," Department of Econometrics, Tilburg University, Tilburg, The Netherlans, 2001.