OPTIMIZATION OF COORDINATE TRANSFORMATION MATRIX FOR $H_{\infty}$ STATIC-OUTPUT-FEEDBACK CONTROL OF LINEAR DISCRETE-TIME SYSTEMS

Zhi-Yong Feng, Li Xu, Jinhua She, and Xue-Xun Guo

ABSTRACT

This paper presents a new iterative algorithm as an upgrade to sufficient LMI conditions for the $H_{\infty}$ static-output-feedback (SOF) control of discrete-time systems. Based on an analysis of the structures of the coordinate transformation matrix and the Lyapunov matrix, the open question of how to fix the Lyapunov matrix structure raised by G. I. Bara and M. Boutayeb is replaced with the question of how to choose the coordinate transformation matrix. Then, an iterative algorithm for selecting the optimum coordinate transformation matrix that produces a locally optimal solution is presented. Finally, numerical examples demonstrate the effectiveness and advantages of this method.

Key Words: $H_{\infty}$ control, static output feedback, linear matrix inequality (LMI).

I. INTRODUCTION

Static-output-feedback (SOF) control is a basic open problem in control theory and its applications. It has attracted considerable attention recently [1–5]. It has been shown that the SOF control problem can be formulated in the form of a biaffine (bilinear) matrix inequality (BMI). However, since a BMI optimization problem is nonconvex and is non-deterministic polynomial-time (NP) hard [6], it is not easy to find a numerical solution. The many attempts that have been made to solve the BMI problem can be categorized into three types. The first is sufficient conditions in the form of a linear matrix inequality (LMI) [7–12]. The second is LMI-based iterative algorithms, including the cone complementarity linearization (CCL) algorithm [13], the min-max algorithm [14], mixed LMI/randomized methods [5], and a combination of convex-concave decompositions and linearization approaches [15]. The third is nonlinear or nonconvex programming methods [16–18].

Programs have recently been developed to solve BMIs by applying various optimization methods, for example, PENBMI [2] and BMIsolver [15].

Among the approaches, sufficient LMI conditions are particularly attractive due to their computational simplicity, and considerable efforts have been constantly made to reduce the cost associated with such simplicity, i.e., the conservativeness of sufficient LMI conditions. For example, a sufficient LMI condition [7] was first devised by forcing a Lyapunov variable to have a two-block diagonal structure. Lee et al. improved on that by using a slack-variable technique in [9]. The structural restriction imposed on Lyapunov variables was bypassed by using two-block diagonal structured slack variables. The conservativeness of this kind of LMI condition lies in the structural restriction on the Lyapunov variable [7] or slack variables [9]. A necessary and sufficient nonlinear condition [8] for the SOF control problem was obtained by extending the result for a continuous-time system in [19] to a discrete-time system. However, the problem of how to fix the Lyapunov matrix structure is difficult. To circumvent this difficulty, another kind of sufficient LMI-based condition [8] was developed. The conservativeness lies in the enlargement of a term in the LMI condition of [8].

A common feature of all the results obtained in [7–9] is that linear time-invariant (LTI) coordinate transformations are performed on the original system to derive sufficient LMI conditions. However, the effects of different choices for those transformation matrices were not explained; and usually certain special choices were used. In [20,21], a parametrization form of the coordinate transformation matrices was presented; and it was pointed out that different...
choices of coordinate transformation matrices affect the feasibility of the LMI conditions. Nevertheless, the basic relationship between the coordinate transformation and the solution of the LMI conditions has not yet been clarified.

Very recently, [11] used the technique of singular-value decomposition (SVD) to derive all applicable LTI coordinate transformation matrices; and they explained the relationship between the coordinate transformation and the feasibility of the LMI conditions. The result in [9] was improved in [11] by using block-triangular structured slack variables instead of two-block diagonal structured ones and by choosing a proper coordinate transformation matrix. Nevertheless, the problem of how to select the coordinate transformation matrix in an algorithmic way remains to be investigated. Furthermore, though it was pointed out in [11] that the question of how to fix the Lyapunov matrix structure raised in [8] is implicitly related to the problem of how to choose the coordinate transformation matrix, further exploration on this point is also required.

This study aimed to reduce or even eliminate the conservativeness of sufficient LMI conditions in [8,9,11] through proper selection of the coordinate transformation matrix. We found that the problem of fixing the Lyapunov matrix structure in [8] can be replaced with the problem of choosing a coordinate transformation matrix. Moreover, an iterative algorithm was devised as an upgrade to the LMI conditions of [8] that chooses the optimum coordinate transformation matrix. The algorithm first finds an initial coordinate transformation matrix. Then, it produces a controller with locally optimal closed-loop $H_{\infty}$ performance by selecting the optimum coordinate transformation matrix. This method can be extended to solve the problems raised in [11,12] of how to choose a coordinate transformation matrix. In addition, it is computationally much simpler than some other iterative LMI methods and nonlinear/nonconvex programming methods.

This paper is organized as follows: Section II gives a description of the problem under consideration and restates some results in [8]. Section III analyzes the effects of different choices of coordinate transformation matrices. Then, an iterative algorithm is presented that solves the $H_{\infty}$ SOF control problem. In Section IV, numerical examples demonstrate the effectiveness and benefits of the method. Finally, some concluding remarks are made in Section V.

**Notation.** Throughout this paper, $\mathbb{C}$ indicates the complex plane; $\mathbb{R}^n$ denotes the $n$-dimensional Euclidean space; $\mathbb{R}^{n \times m}$ is the set of all $n \times m$ real matrices; and $I_n$ is an $n \times n$ identity matrix. For a real matrix $P$, $P > 0$ ($< 0$) means that $P$ is real symmetric and positive (negative) definite. In any matrix, $*$ denotes symmetric terms.

## II. PROBLEM FORMULATION AND PRELIMINARIES

Consider the discrete-time system

\[
\begin{align*}
\begin{bmatrix}
\dot{x}(k+1) = & A x(k) + B_w w(k) + B_u u(k) \\
\dot{z}(k) = & C_x x(k) + D_{wz} w(k) + D_{uz} u(k) \\
y(k) = & C_z x(k)
\end{bmatrix}
\end{align*}
\]  

(1)

where $x(k) \in \mathbb{R}^n$ is the system state, $w(k) \in \mathbb{R}^q$ is the exogenous disturbance input, $u(k) \in \mathbb{R}^q$ is the control input, $z(k) \in \mathbb{R}^p$ is the controlled output, and $y(k) \in \mathbb{R}^l$ is the measured output. $A$, $B_w$, $B_u$, $C_x$, $C_z$, $D_{wz}$, and $D_{uz}$ are constant matrices with appropriate dimensions. Without loss of generality, we assume that $C_z$ has full row rank.

Let $T_r$ be a nonsingular matrix such that

\[
C_r T_r^{-1} = [I_l \ 0]
\]  

(2)

Using the matrix $T_r$ to perform a coordinate transformation on System (1), i.e., $\bar{z}(k) = T_r z(k)$, yields the algebraically equivalent system

\[
\begin{align*}
\begin{bmatrix}
\dot{\bar{z}}(k+1) = & \bar{A} \bar{z}(k) + \bar{B}_w w(k) + \bar{B}_u u(k) \\
\bar{z}(k) = & \bar{C}_x \bar{z}(k) + D_{wz} w(k) + D_{uz} u(k) \\
\bar{y}(k) = & \bar{C}_y \bar{z}(k)
\end{bmatrix}
\end{align*}
\]  

(3)

where

\[
\begin{align*}
\bar{A} = & T_r A T_r^{-1}, \quad \bar{B}_w = T_r B_w, \quad \bar{B}_u = T_r B_u \\
\bar{C}_x = & C_r T_r^{-1}, \quad \bar{C}_y = [I_l \ 0]
\end{align*}
\]  

(4)

An SOF control law is defined as

\[
u(k) = Ky(k)
\]  

(5)

where $K \in \mathbb{R}^{qd}$ is a gain matrix to be determined.

Applying the SOF controller (5) to System (3) yields the closed-loop system

\[
\begin{align*}
\begin{bmatrix}
\dot{\bar{z}}(k+1) = & \bar{A} \bar{z}(k) + \bar{B}_w w(k) \\
\bar{z}(k) = & \bar{C}_x \bar{z}(k) + D_{wz} w(k)
\end{bmatrix}
\end{align*}
\]  

(6)

where

\[
\begin{align*}
\bar{A}_z = & \bar{A} + \bar{B}_w K \bar{C}_y, \quad \bar{C}_{iz} = \bar{C}_z + D_{uz} K \bar{C}_y
\end{align*}
\]  

(7)

First, we restate some results from [8].

Let $T_{wz}$ be the closed-loop transfer function from $w$ to $z$. From the bounded real lemma [22], we know that the closed-loop system (6) is stable and that the $H_{\infty}$ norm of the corresponding closed-loop transfer function, $T_{wz}$, is smaller than $\gamma$ (that is, $||T_{wz}||_{\infty} < \gamma$) if and only if there exist a positive definite matrix $P \in \mathbb{R}^{n \times n}$ and a control gain matrix $K \in \mathbb{R}^{qd}$ satisfying the BMI

\[

\text{BMI}
\]
The sufficient condition below circumvents, with the cost of conservativeness, both the nonlinearity problem in Lemma 1 and the problem of fixing the Lyapunov matrix structure.

**Lemma 2.** Theorem 4.2 of [8]. For a given positive scalar $\gamma$, System (1) is stabilizable by an SOF controller (5), and the $H_\infty$-norm of the corresponding closed-loop transfer function, $T_{wv}$, is smaller than $\gamma$ (that is, $\|T_{wv}\|_\infty < \gamma$) if there exist a positive definite matrix $P \in \mathbb{R}_+^{n \times n}$ and a scalar $\delta$ such that the following LMIs hold:

$$
\begin{align*}
-P + \delta \bar{B}_w \bar{B}_w^T A P & \preceq \delta \bar{B}_w D_{tw} \bar{B}_w^T + \bar{B}_w^T \bar{B}_w < 0 \\
* & \quad -P C_z^T P C_z^T < 0 \\
* & \quad -\gamma I < 0
\end{align*}
$$

where $\bar{A}$, $\bar{B}_w$, and $\bar{C}_z$ are defined in (4). Note that $P_{22} \in \mathbb{R}_+^{n_{w} \times n_{w}}$ is the (2, 2)-block of $P$.

Ignoring $H_\infty$ performance in Lemma 2 gives the following lemma:

**Lemma 3.** Theorem 3.3 of [8]. System (1) with $w(k) = 0$ is stabilizable by an SOF controller (5) if there exist a positive definite matrix $P \in \mathbb{R}_+^{n \times n}$ and a scalar $\delta$ such that the following LMIs hold

$$
\begin{align*}
-P + \delta \bar{B}_w \bar{B}_w^T A P & \preceq \delta \bar{B}_w D_{tw} \bar{B}_w^T + \bar{B}_w^T \bar{B}_w < 0 \\
* & \quad -P C_z^T P C_z^T < 0 \\
* & \quad -\gamma I < 0
\end{align*}
$$

where $\bar{A}$, $\bar{B}_w$, and $\bar{C}_z$ are defined in (4).

**Remark 2.** Since Lemma 2 is an extension of Lemma 3 to the $H_\infty$ control case, it is easy to see from the proof of Theorem 3.3 in [8] that Lemma 2 will be equivalent to Lemma 1 if we replace $P_{22} - P_{12}^T P_{11}^{-1} P_{21}$ with $P_{22}$ in (13). Although the enlargement of $P_{22} - P_{12}^T P_{11}^{-1} P_{21}$ to $P_{22}$ leads to the conservativeness, it results in a tractable LMI condition. This study aims to reduce the conservativeness caused by this replacement.

**Remark 3.** Remark 3.3 of [8] pointed out that condition (14) means that the pair $(\bar{A}, \bar{B}_w)$ is stabilizable, and condition (15) means that $\bar{A}_{32}$ is stable, where $\bar{A}_{32}$ is the (2, 2)-block of $\bar{A}$ according to the partition of $P$. So, a necessary condition for the existence of a solution to the LMIs of Lemma 3 is

$$K = RP_1^{-1}$$

**Remark 1.** The condition (11) is nonlinear due to the nonlinear terms ($T_{N}^{-1}$, $T_{N} P_{d}$, etc.). So, it is not tractable by convex optimization techniques. When the matrix $N$ is fixed, that is, $N$ has a particular structure, inequality (11) becomes linear in $P_{d}$ and $P_{2}$. Thus, it is easy to solve numerically. The matrix $T_{N}$ acts as a similarity (coordinate) transformation for the closed-loop system, and this transformation yields a state-space representation that admits a block-diagonal Lyapunov function [8]. The open problem in [8] is “how to fix the Lyapunov matrix structure such that one can determine the existence of an SOF gain”.

The condition (11) is nonlinear due to the nonlinear terms ($T_{N}^{-1}$, $T_{N} P_{d}$, etc.). So, it is not tractable by convex optimization techniques. When the matrix $N$ is fixed, that is, $N$ has a particular structure, inequality (11) becomes linear in $P_{d}$ and $P_{2}$. Thus, it is easy to solve numerically. The matrix $T_{N}$ acts as a similarity (coordinate) transformation for the closed-loop system, and this transformation yields a state-space representation that admits a block-diagonal Lyapunov function [8]. The open problem in [8] is “how to fix the Lyapunov matrix structure such that one can determine the existence of an SOF gain”.

Z. Y. Feng et al.: Optimization of Coordinate Transformation Matrix for $H_\infty$ SOF Control

© 2014 Chinese Automatic Control Society and Wiley Publishing Asia Pty Ltd
**III. MAIN RESULTS**

By analyzing the structure of the coordinate transformation matrix by means of the SVD factorization, we show that the open question of how to fix the Lyapunov matrix structure raised in [8] can be replaced with the question of how to choose the coordinate transformation matrix $T_y$. Then, we present a strategy for iteratively updating $T_y$ that produces a locally optimal solution based on the LMI conditions of [8]. This section presents an iterative algorithm for this strategy.

3.1 All applicable LTI coordinate transformation matrices

Since $C_y \in \mathbb{R}^{n \times m}$ has full row rank, the SVD factorization of $C_y$ is

$$C_y = U_y [C_0 \ 0] V_y^T$$

where $U_y \in \mathbb{R}^{n \times n}$ and $V_y \in \mathbb{R}^{m \times m}$ are unitary matrices, and $C_0 \in \mathbb{R}^{n \times d}$ is a diagonal matrix with positive diagonal elements in decreasing order.

The following lemma gives all the applicable LTI coordinate transformation matrices.

**Lemma 4** [11]. For a given $C_y \in \mathbb{R}^{n \times m}$ with full row rank (that is, rank $C_y = l$) and its SVD factorization given by (17), there always exists a nonsingular matrix $T_y \in \mathbb{R}^{m \times m}$ such that $C_y T_y^{-1} = [I_l \ 0]$ and

$$T_y^{-1} = V_y \begin{bmatrix} (U_y C_0)^{-1} & 0 \\ T_1 & T_{22} \end{bmatrix}$$

where $T_{22} \in \mathbb{R}^{(m-n) \times (m-n)}$ is a constant matrix and $T_{22} \in \mathbb{R}^{(n-b_0a-n)}$ is a nonsingular constant matrix.

**Remark 4.** Based on the SVD factorization of $C_y$, the orthonormal basis for the orthogonal complement and the Moore-Penrose pseudoinverse of matrix $C_y$ can be written as

$$C_y^+ = V_y \begin{bmatrix} 0 & I_{m-l} \\ I_{n-l} & 0 \end{bmatrix} C_y^T (C_y C_y^T)^{-1} = V_y \begin{bmatrix} (U_y C_0)^{-1} & 0 \\ 0 & I \end{bmatrix}$$

An equivalent parametrized form of (18) is

$$T_y^{-1} = \begin{bmatrix} C_y^+ + C_y^+ T_{21} & C_y^+ T_{22} \\ C_y^+ & C_y^+ \end{bmatrix}$$

where $T_{21}$ was taken to be $\begin{bmatrix} C_y^+ & C_y^+ \end{bmatrix}$ in [9,10] (note that $[C_y^+ C_y^+]^T = [C_y^+ C_y^+]$). It is a special case of (18) and (19).

**Remark 5.** The parametrization form (19) is dual to that given in [20,21]. Since the parametrization was given in those references without proof, it is not clear whether or not the parametrization contains all the applicable LTI coordinate transformation matrices. In contrast, the proof of Lemma 4 in [11] shows that the parametrized form (18), or equivalently (19), presents all applicable LTI coordinate transformation matrices. In this paper, we focus on how to select an optimal coordinate transformation matrix to solve the $H_{\infty}$ SOF control problem. So, it is important to specify the range of applicable coordinate transformation matrices.

For a linear time-invariant state-space system, the transformed system is algebraically equivalent to the original one for any nonsingular state coordinate transformation. However, since LMIs (12) and (13) are only sufficient for BMI (8), the feasibility of those LMIs is affected by the choice of $T_y$. The following theorem specifies how $T_y$ affects the feasibility.

**Theorem 1.** The choice of the nonsingular matrix $T_{22}$ of $T_y^{-1}$ does not affect the feasibility of (12) and (13), but the choice of $T_{21}$ does affect it.

**Proof.** Substituting (4) into (12), and pre- and post-multiplying (12) by $\text{diag} \{ T_y^{-1}, T_y^{-1} I, I \}$ and $\text{diag} \{ T_y^{-T}, T_y^{-T} I, I \}$ yield

$$\begin{bmatrix} -T_y^{-1} P_T T_y^{-1} + B_y D_y B_y^T & A T_y^{-1} P_T T_y^{-1} & \delta B_y D_y B_y^T & B_{22} \\ * & -T_y^{-1} P_T T_y^{-1} & T_y^{-1} P_T C_y^+ & 0 \\ * & * & -\gamma I + \delta D_y D_y^T & D_{22} \\ * & * & * & -\gamma I \end{bmatrix} < 0 \quad (20)$$

Substituting (4) into (13) and pre- and post-multiplying (13) by $\text{diag} \{ T_y^{-1}, I, I \}$ and $\text{diag} \{ T_y^{-T}, I, I \}$ give

$$\begin{bmatrix} -T_y^{-1} P_T T_y^{-1} & 0 & B_{22} \\ * & -\gamma I & D_{22} \\ * & * & -\gamma I \end{bmatrix} + \begin{bmatrix} A & 0 \\ C_y & 0 \end{bmatrix} \begin{bmatrix} T_y^{-1} I & 0 \\ 0 & P_{22} \end{bmatrix} \begin{bmatrix} A^T \\ C_y \end{bmatrix} < 0 \quad (21)$$
From the definition of $T_y^{-1}$, we have

$$T_y^{-1}P_{T_y^{-T}} = V_y \left[ \begin{array}{cc} (U_c C_o)^{-1} & 0 \\ T_{21} & T_{22} \end{array} \right] P_{T_y} V_y^T$$

$$= V_y \left[ \begin{array}{cc} (U_c C_o)^{-1} P_{11} (U_c C_o)^{-1} T_{21} + P_{12} T_{22} \end{array} \right] V_y^T$$

and

$$T_y^{-1}P_{T_y^{-T}} = V_y \left[ \begin{array}{cc} 0 & 0 \\ 0 & P_{T_y} \end{array} \right] V_y^T$$

Next, we show that the choice of $T_{21}$ affects the feasibility of LMI (12) and (13). If we let

$$Q_{11} = P_{11}$$

$$Q_{12} = P_{12} T_{21}^T + P_{22} T_{22}^T$$

$$Q_{22} = T_{12} P_{T_y} T_{22}^T + T_{22} P_{T_y} T_{22}^T$$

$$= T_{21} Q_{21} + Q_{12} T_{21}^T - T_{21} Q_{11} T_{21}^T + T_{22} P_{T_y} T_{22}^T,$$

then we can rewrite (22) as

$$T_y^{-1}P_{T_y^{-T}} = V_y \left[ \begin{array}{cc} (U_c C_o)^{-1} Q_{21} (U_c C_o)^{-1} Q_{12} \end{array} \right] V_y^T$$

and we can rewrite (23) as

$$T_y^{-1}P_{T_y^{-T}} = V_y \left[ \begin{array}{cc} 0 & 0 \\ 0 & Q_{22} - T_{21} Q_{21} - Q_{12} T_{21}^T + T_{22} Q_{11} T_{21}^T \end{array} \right] V_y^T$$

Substituting (24) and (25) into (20) and (21) shows that the choice of $T_{21}$ does affect the feasibility of LMI (12) and (13).

**Remark 6.** How the choice of the orthogonal complement of $B_c$ in [20] (or $B$ in [21]), which plays a similar role as $T_{22}$, affects the solvability of the corresponding LMI was an unsolved problem in [20,21]. However, using a procedure similar to that in the proof of Theorem 1, it is easy to show that a different choice for the orthogonal complement of $B_c$ (or $B$) does not affect the solvability of the LMI in [20,21].

Now, we have the problem of how to choose an appropriate $T_{21}$ to reduce the conservativeness of LMI (12) and (13). In order to establish a strategy to solve this problem, we first analyze the relationship between the problem of how to choose $T_{21}$ and the problem of how to choose $N$, which is an open problem raised in [8] as mentioned previously.

As mentioned in Remark 1, $T_y$ can be viewed as a coordinate transformation matrix for System (3). Thus, we can treat $T_y^* T_y$ as a coordinate transformation matrix for the original system (1). Using (10) and (18), and choosing $T_{22}$ to be the identity matrix give

$$T_y^{-1} T_y = V_y \left[ \begin{array}{cc} (U_c C_o)^{-1} & 0 \\ T_{21} & I \end{array} \right] = V_y \left[ \begin{array}{cc} (U_c C_o)^{-1} & 0 \\ T_{21} + N^T I \end{array} \right].$$

The problem of choosing the coordinate transformation matrix was not addressed in [8]. How to choose the matrix $N$ was raised as an open problem. It is clear from (26) that $T_{21}$ and $N$ play the same role in the condition in Lemma 1. We can fix either $T_{21}$ or $N$ (especially as a zero matrix) and just choose the other. However, since $N$ is related to the structure of the Lyapunov matrix $P$, how to choose $N$ is a difficult problem. So, we choose $T_{21}$ in this study.

### 3.2 Solution of $H_\infty$ SOF control

As explained in Remark 2, the replacement of $P_{T_y} = P_{T_1} P_{T_1} P_{T_1} P_{T_2}$ in (13) leads to the conservativeness of Lemma 2. If we can find a coordinate transformation matrix $T_y$ (or more specifically, $T_{21}$) that results in a block-diagonal solution in $P$ in Lemma 2 (that is, $P_{12} = 0$), then $P_{T_y} = P_{T_1} P_{T_1} P_{T_2}$ automatically becomes zero. This reduces the conservativeness of Lemma 2. Note that the solution is only locally optimal since $T_{21}$ may not be unique. So, the selection of $N$ in Lemma 1 is replaced with the selection of $T_{21}$ in Lemma 2. For a properly selected $T_{21}$, $N = P_{T_1} P_{T_2} = 0$. That gives us a hint about how to choose the optimum $T_{21}$ by recalling (26).

Suppose that a feasible solution, $P$, for the optimization problem (16) is obtained for a selected initial coordinate transformation matrix $T_y$ with $T_{21} = T_{21}^{(0)}$. We calculate $N = P_{T_1} P_{T_2}$, set $T_{21} = T_{21}^{(0)} + N$, and update $T_y$ by setting $T_{21} = T_{21}^{(0)}$. Then, we can use the new coordinate transformation matrix, $T_y$, to solve problem (16) again. We repeat these steps until a stop criterion is satisfied. This procedure produces a locally optimal coordinate transformation matrix, $T_y$, and also a locally minimum $\gamma$. An analysis of the convergence of the iterative strategy is given in the next subsection.

Now, the problem becomes how to determine the initial coordinate transformation matrix $T_y$ (or more specifically, $T_{21}^{(0)}$). Since a given choice of $T_y$, as in [9,10], does not always ensure the existence of a feasible solution for the LMI
conditions (12) and (13), a strategy for the selection of the initial coordinate transformation matrix is needed.

We define
\[
A = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} = T AT^{-1}, \quad T = \begin{bmatrix} U, C_0 & 0 \\ 0 & I \end{bmatrix} V_c^T. \tag{27}
\]

The theorem below is useful for selecting an initial matrix for \(T_{21}\).

**Theorem 2.** If LMI (13) (or LMI (15)) is feasible, then \(\hat{A}_{22} - T_{21} \hat{A}_{12}\) is Schur stable.

**Proof.** From the definitions of \(T_y, T, A, \) and \(\hat{A}\), we have
\[
\bar{A} = T_y A T_y^{-1} = \begin{bmatrix} U, C_0 & 0 \\ -T_{22} T_{12} U, C_0 T_{22} \\ T_{21} & T_{22} \end{bmatrix} V_c^T A V_c (U, C_0)^{-1} \begin{bmatrix} 0 & T_{21} & T_{22} \end{bmatrix} \tag{28}
\]
and
\[
\bar{A}_{22} = T_{21}^{-1} (\hat{A}_{22} - T_{21} \hat{A}_{12}) T_{22} \tag{29}
\]

\(\bar{A}_{22}\) is stable if and only if \(\hat{A}_{22} - T_{21} \hat{A}_{12}\) is Schur stable. From Remark 3, we see that the condition that \(\bar{A}_{22}\) be stable is a necessary condition for the feasibility of LMI (13) in Lemma 2 and also for the feasibility of LMI (15) in Lemma 3. This completes the proof.

**Remark 7.** \(\hat{A}_{22} - T_{21} \hat{A}_{12}\) is Schur stable if and only if the pair \((\hat{A}_{22}, - \hat{A}_{12})\) is detectable, or equivalently, the pair \((\hat{A}_{22} - \hat{A}_{12}', \hat{A}_{12}')\) is stabilizable. It is easy to see that \((\hat{A}_{22}, - \hat{A}_{12}')\) is stabilizable if and only if \((\hat{A}_{22} - \hat{A}_{12}')\) is stabilizable. Therefore, according to Remark 3, a necessary condition for the SOF-stabilizability of System (1) is that the pair \((\hat{A}_{22} - \hat{A}_{12}')\) be stabilizable. So, a necessary condition for the feasibility of the conditions in Lemmas 1–3 is that \((A, B)\) and \((\hat{A}_{22} - \hat{A}_{12}')\) both be stabilizable.

Combining the above lemmas and theorems, we developed the algorithm below to solve the \(H_{\infty}\) SOF control problem. In the algorithm, \(m\) and \(k\) are iteration numbers, \((n - l)\) is the size of \(\hat{A}_{22}\), \(m_{\max}\) and \(k_{\max}\) are given maximum numbers of iterations, and \(\epsilon\) is a prescribed tolerance.

**Algorithm 1 (Design of \(H_{\infty}\) SOF controller).**

Step 1. Check if \((A, B)\) is stabilizable. If it is, then continue. Otherwise, stop.
Step 2. Check if \((\hat{A}_{22} - \hat{A}_{12}')\) is stabilizable. If it is, then continue. Otherwise, stop.
Step 3. Set \(m = 0\). Compute the eigenvalues of the uncontrollable modes in \((\hat{A}_{22} - \hat{A}_{12}')\). Let \(n_u\) be the number of those uncontrollable modes. Set \(n_u = 0\) if \((\hat{A}_{22} - \hat{A}_{12}')\) is completely controllable.
Step 4. Choose \((n - l - n_u)\) complex numbers randomly in the unit circle in \(C\). Set the desired closed-loop poles to the \(n_u\) eigenvalues of the uncontrollable modes and the \((n - l - n_u)\) complex numbers.
Step 5. Use a pole placement algorithm to calculate a feedback gain matrix, \(F\), such that the eigenvalues of \((\hat{A}_{22} - \hat{A}_{12}')F\) are the \((n - l)\) poles given in Step 4. Note that if \(\hat{A}_{22} = 0\), then \(n_u = n - l \) and \(F\) can be chosen to be any appropriately dimensioned matrix. Set \(T_{21} = F^T\), and \(T_{22} = I\) in (18) to obtain a \(T_y\). If (14) and (15) are feasible with this \(T_y\), then set \(k = 0\) and \(T_{21} = F^T\) and continue. Otherwise, if \(m < m_{\max}\), then set \(m = m + 1\) and go to Step 4; otherwise stop.
Step 6. Use \(T_{21} = T_{21}^{(0)}\) to solve problem (16) for \(P, \delta\), and \(\gamma\). Calculate \(N = P_1^{-T} P_2\). If \(|N| < \epsilon\) or \(k > k_{\max}\), continue. Otherwise, set \(T_{21}^{(i+1)} = T_{21}^{(i)} + N^T\) and \(k = k + 1\) and repeat Step 6.
Step 7. Use the \(T_y\) and \(N\) obtained in Step 6 to solve inequality (11) for \(P_1, P_2\), and \(R\); and calculate the \(H_{\infty}\) SOF control gain, \(K = RP_1^{-1}\).

**Remark 8.** The algorithm can be divided into two phases: stabilization and optimization. In the stabilization phase, if either \((A, B)\) or \((\hat{A}_{22} - \hat{A}_{12}')\) is not stabilizable, System (1) is not \(SOF\)-stabilizable. In Step 5, a pole placement method is used to calculate possible candidates for an initial matrix for \(T_{21}^{(0)}\). If there exists a feasible stabilization solution for LMIs (14) and (15), then the initial matrix \(T_{21}^{(0)}\) is found, and the algorithm proceeds to the optimization phase. In the optimization phase, once an initial matrix for \(T_{21}^{(0)}\) has been found that ensures the feasibility of (14) and (15), a locally optimal \(T_{21}\) is obtained by iteratively adjusting \(T_{21}\). In the final step, this \(T_{21}\) is used to compute the \(H_{\infty}\) SOF control gain, \(K\).

### 3.3 Discussion on the algorithm

This subsection explores the features of the algorithm.

#### 3.3.1 Choice of \(T_{21}^{(0)}\)

Since the SOF control problem is nonconvex, different initial choice of \(T_{21}\) may result in different local optimum. So, the initial choice of \(T_{21}\) is an important factor that determines the control performance. In Step 5 of Algorithm 1, an \(F^T\) that places the desired closed-loop poles for the pair \((\hat{A}_{22} - \hat{A}_{12}')\) is used as a possible initial choice for \(T_{21}\). We now introduce a treatment that allows the possibilities for the selection of \(T_{21}^{(0)}\) to be explored.
We randomly generate a nonsingular matrix, $T_*$, and compute a feedback gain matrix, $F$, to stabilize the pair $(T_*A_*^{-1}T_*^T, -T_*A_*^{-1})$ for desired closed-loop poles. This yields the feedback gain matrix

$$
F = F_*T_*,
$$

(30)

Clearly, $F$ depends on the choice of $T_*$. 3.3.2 Convergence analysis

Suppose $\gamma^{(k)}, \delta^{(k)},$ and $P^{(k)} = \begin{bmatrix} P^{(k)}_{11} & P^{(k)}_{12} \\ P^{(k)}_{21} & P^{(k)}_{22} \end{bmatrix}$ are a solution of (16) at the $k$th iteration in Step 6 of Algorithm 1, where the coordinate transformation matrix is $T^{(k)}$. First, we show that the sequence $\gamma^{(k)}$ decreases monotonically.

Let $N^{(k)} = (P^{(k)}_{11})^T P^{(k)}_{12}$ and $T^{(k+1)} = T^{(k)} + [N^{(k)}]_T$. Then,

$$
(T^{(k)})^{-1} P^{(k)} (T^{(k)})^{-T} = (T^{(k+1)})^{-1} P^{(k)} (T^{(k)})^{-T}
$$

(31)

where $P^{(k)}_{11} = P^{(k)}_{11},$ and $P^{(k)}_{22} = P^{(k)}_{22} - N^{(k)}_2 P^{(k)}_{12} N^{(k)}_1$. It is clear from (31) that (12) and (13) are feasible for $T^{(k+1)}$ and $\gamma = \gamma^{(k)}$ with the solution $\delta = \delta^{(k)}$ and $P = \begin{bmatrix} P^{(k)}_{11} & 0 \\ 0 & P^{(k)}_{22} \end{bmatrix}$. Therefore, in the $(k+1)$th iteration, $\gamma^{(k+1)}$ is less than or equal to $\gamma^{(k)}$ because $P^{(k+1)}$ is not restricted to be block-diagonal.

So, the sequence $\gamma^{(k)}$ monotonically decreases; and when $N^{(k)}$ (or $P^{(k)}_{12}$) approaches zero, $\gamma^{(k)}$ approaches a local minimum. Numerical verification showed that, while the sequence $[N^{(k)}]$ converged to a small enough value in most cases, there were a few nonconvergent cases. In those cases, $\gamma^{(k)}$ converged to a local optimal value, but $\|N^{(k)}\|$ varied in a small range. It is worth mentioning that, no matter whether $\|N^{(k)}\|$ converges or not, the convergence of $\gamma^{(k)}$ is always guaranteed.

3.3.3 Dual solutions

Consider the discrete-time system

$$
\begin{align*}
\dot{x}(k+1) &= Ax(k) + Bu(k) + B_*w(k) \\
\dot{z}(k) &= C_*x(k) + D_*w(k) \\
\gamma(k) &= C_*x(k) + D_*w(k)
\end{align*}
$$

(32)

Without loss of generality, we assume that $B_*$ has full column rank. Let $T_*$ be a nonsingular matrix such that

$$
T_*B_* = [I, \ 0]^T
$$

(33)

It is easy to derive dual results for System (32) using $T_*$ as a coordinate transformation matrix. Those results are straightforward and are omitted for brevity.

3.3.4 Extension of Algorithm 1 to other LMI conditions

Algorithm 1 was derived based on the LMI conditions in [8]. However, the idea used in the algorithm can also be used to reduce the conservativeness of other LMI conditions. For example, the LMI conditions in [9,11] restrict the slack variables to two-block diagonal or block-triangular structures. This makes the methods conservative. Following the same idea as that of Algorithm 1, we can update the LMI conditions in [9,11] by iteratively adjusting $T_{21}$. When a locally optimal $T_{21}$ is obtained, the solution for the Lyapunov or slack variables can be automatically two-block diagonal. This reduces the conservativeness. On the other hand, the presented iterative procedure is not applicable to the LMI conditions in [7,21] because the Lyapunov variables are forced to be two-block diagonal. This makes it impossible to use $N = P^{(1)}_{11}P^{(1)}_{12}$ to update $T_{21}$.

3.3.5 Advantages and disadvantages of our method

Optimizing the choice of $T_*$ greatly reduces the conservativeness of the LMI conditions in [8,9,11].

A common feature of most iterative algorithms for solving the $H_\infty$ SOF control problem (CCL algorithm [13], iterative LMI (ILMI) method [23], method in [24], etc.) is that $\gamma$ is fixed. In contrast, our method finds a minimum $\gamma$.

Unlike those nonlinear or nonconvex programming methods, our method does not use any sophisticated nonlinear optimization techniques. So, it is simple to use and easy to extend to the solution of other control problems, such as decentralized $H_\infty$ SOF control, simultaneous $H_\infty$ SOF control [9], the $H_\infty$ SOF control of 2D discrete systems [12], etc.

A drawback of our method is that it cannot deal with a standard generalized plant. We assume that either $D_{uw}$ or $D_{wu}$ is zero.

IV. NUMERICAL EXAMPLES

The numerical examples in this section demonstrate the effectiveness and efficiency of our method and compare it to other recently devised methods. The LMI toolbox of MATLAB was used to solve the LMI conditions in [8–10] for the examples.

PENBMI [2], HIFOO [4] are available for solving the $H_\infty$ SOF control problem. PENBMI is a commercially available program that locally solves optimization problems with a quadratic objective and BMI constraints. A free developer license is available to academic users. HIFOO is a freeware...
program that solves fixed-order stabilization and performance optimization problems. It has been extended to the control of discrete-time systems [4]. The numerical examples below compare our method with PENBMI (Version 2.1) and HIFOO-D (HIFOO for discrete-time systems). Both PENBMI and HIFOO-D were run in the default mode. Note that $\varepsilon$ was set to 0.00001 and $k_{\max}$ was set to 800 in Step 6 of Algorithm 1.

**Example 1.** $H_\infty$ SOF control: Consider the unstable plant [9,10]

$$x(k+1) = \begin{bmatrix} 3.0 & 0.3 & 2 \\ 0.3 & 0.6 & -0.6 \\ 1 & 0 & 1 \end{bmatrix} x(k) \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} w(k) + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} x(k)$$

where $\alpha$ is either 0 or 1.

Table I lists numerical results obtained by various methods. $K$ is the $H_\infty$ SOF control gain, $\gamma_{\min}$ is the minimum $\gamma$ obtained, and $\gamma$ is the actual $H_\infty$-norm that $K$ produces.

Setting $T_{22} = I$ and letting $T_{21} = [t_1 \ t_2]^T$ yield $T_{y^{-1}}$ from Lemma 4:

$$T_{y^{-1}} = \begin{bmatrix} 0.7071 & -0.7071 & 0 \\ 0.7071 & 0.7071 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.7071 & 0 & 0 \\ t_1 & 1 & 0 \\ t_2 & 0 & 1 \end{bmatrix}$$

When $T_{y}$ was chosen to be the one in [8–10], the results were conservative for $\alpha = 1$; and a feasible solution could not be found for $\alpha = 0$. In contrast, when we used Algorithm 1 to optimize the choice of $T_{y}$, we obtained much better results. We ran Algorithm 1 many times and found that the locally optimal $T_{21}$ was always $\begin{bmatrix} -0.3295 \\ \vdots \end{bmatrix}$ for $\alpha = 1$, and $\begin{bmatrix} -0.3267 \\ \vdots \end{bmatrix}$ for $\alpha = 0$ even though the initial $T_{21}$ was different every time. This indicates that this example may have only one optimum, that is, the solution might be a global optimum.

Note that, if we use the $T_{y}$ produced by Algorithm 1 to solve the LMI conditions in [7,9,11], we get the same results as those obtained by using Algorithm 1. We can also get the same results by substituting the LMI conditions in [9,11] for the LMIs in Step 6 of Algorithm 1. However, since the LMIs in [8] contain fewer variables than those in [9,11], they produce results more quickly.

Clearly, PENBMI, HIFOO-D, and Algorithm 1 give the same results for this example. Next, we compared the performance of these three methods on 100 randomly generated numerical examples.

<table>
<thead>
<tr>
<th>$\alpha = 1$</th>
<th>$\gamma_{\min}$</th>
<th>$\gamma$</th>
<th>$\alpha = 0$</th>
<th>$\gamma_{\min}$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td></td>
<td></td>
<td>$K$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-0.3760]$</td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>[8]</td>
<td>$[-0.9698]$</td>
<td>11.2797</td>
<td>9.7868</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$[-0.1775]$</td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>[10]</td>
<td>$[-0.9866]$</td>
<td>10.1240</td>
<td>9.1140</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$[-0.1907]$</td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>$[-0.2369]$</td>
<td></td>
<td></td>
<td>$0.5890$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-0.2369]$</td>
<td></td>
<td></td>
<td>$0.5890$</td>
<td></td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>$[-1.0058]$</td>
<td>8.5724</td>
<td>8.5724</td>
<td>$[-0.9682]$</td>
<td>29.0734</td>
</tr>
<tr>
<td></td>
<td>$[-0.2369]$</td>
<td></td>
<td></td>
<td>$0.5890$</td>
<td></td>
</tr>
</tbody>
</table>

© 2014 Chinese Automatic Control Society and Wiley Publishing Asia Pty Ltd
Example 2. We randomly generated 100 numerical examples for System (1) with the dimensions $n = 4$, $q = 4$, $r = 4$, $p = 3$, and $l = 2$.

We first compare the minimum $\gamma$ produced by the three methods. It is observed that, for many examples, the systems have different local optima. Since both HIFOO-D and Algorithm 1 start from randomly selected points, they may give different results between runs. We ran each program five times for every example and chose the best result. HIFOO-D and Algorithm 1 worked well for all 100 examples, but PENBMI failed in 67 cases. In the 33 cases where PENBMI succeeded, HIFOO-D and Algorithm 1 gave nearly the same or better results than PENBMI did. (When values for the minimum $\gamma$ agree within 2%, they are regarded as nearly the same.) Reference [2] mentions that PENBMI may suffer from numerical problems related to poor conditioning or bad data scaling. Our numerical tests showed that the failure probability of PENBMI is very high for $H_\infty$ SOF control problems.

Then, we specifically compare HIFOO-D and Algorithm 1. Fig. 1 shows a bar diagram of $\log_2$ of the ratio of the minimum $\gamma$s produced by the two methods. Positive values mean that HIFOO-D is better, and negative values mean that Algorithm 1 is better. These two methods give almost the same minimum $\gamma$ in 80 cases; while HIFOO-D is significantly better in 14 cases, and Algorithm 1 is significantly better in six cases. Fig. 2 shows a bar diagram of $\log_{10}$ of the ratio of the CPU times for the two methods. Positive values mean that HIFOO-D is faster, and negative values mean that Algorithm 1 is faster. Algorithm 1 ran faster in 93 cases, and HIFOO-D ran faster in only seven cases.

These numerical examples demonstrate that our algorithm is efficient and effective compared to PENBMI and HIFOO-D. It provides an alternative solution to the $H_\infty$ SOF control problem of discrete-time systems.

V. CONCLUSION

This paper analyzes the relationship between the two open questions raised in [8] and [11]. It shows that the open question of how to fix the Lyapunov matrix structure can be replaced with the question of how to choose the coordinate transformation matrix. Then, an iterative strategy is explained for solving the problem of selecting the coordinate transformation matrix. Moreover, an algorithm is presented for solving the $H_\infty$ SOF control problem for linear discrete-time systems. Numerical examples demonstrate the effectiveness and advantages of the method.

REFERENCES


**Xue-Xun Guo** is Professor at the School of Automotive Engineering, Wuhan University of Technology, Wuhan, China. He is a committee member of SAE-China SUV Branch, and a committee member of China Tractor Society. His current research interests include vehicle dynamics and control, simulation and control of transmission system and braking system, energy-regenerative shock absorber, hydraulic retarder and torque converter.

**Jinhua She** received the B.S. degree in engineering from Central South University, Changsha, China in 1983, and the M.S. and Ph.D. degrees in engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1990 and 1993, respectively. In 1993, he joined the Department of Mechatronics, School of Engineering, Tokyo University of Technology, Tokyo, and in April 2008, he transferred to the University’s School of Computer Science, where he is currently Professor. His current research interests include the application of control theory, repetitive control, process control, Internet-based engineering education, and robotics. Dr She received the IFAC control engineering practice paper prize in 1999 (together with M. Wu and M. Nakano).