Linear Algebra and its Applications 623 (2021) 258–281

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

Minimal-norm static feedbacks using dissipative Hamiltonian matrices

Nicolas Gillis [a,*,1], Punit Sharma [b,2]

[a] Department of Mathematics and Operational Research, University of Mons, Rue de Houdain 9, 7000 Mons, Belgium
[b] Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110016, India

A R T I C L E I N F O

Article history:
Received 16 July 2019
Accepted 4 February 2020
Available online 10 February 2020
Submitted by D. Kressner

MSC:
93B40
34A30
65K05
93D99

Keywords:
Dissipative Hamiltonian system
Static-state feedback
Static-output feedback
Semidefinite optimization

A B S T R A C T

In this paper, we characterize the set of static-state feedbacks that stabilize a given continuous linear-time invariant system pair using dissipative Hamiltonian matrices. This characterization results in a parametrization of feedbacks in terms of skew-symmetric and symmetric positive semidefinite matrices, and leads to a semidefinite program that computes a static-state stabilizing feedback. This characterization also allows us to propose an algorithm that computes minimal-norm static feedbacks. The theoretical results extend to the static-output feedback (SOF) problem, and we also propose an algorithm to compute the minimal-norm SOF. We illustrate the effectiveness of our algorithm compared to state-of-the-art methods for the SOF problem on numerous numerical examples from the COMPLeIB library.

* Corresponding author.
E-mail addresses: nicolas.gillis@umons.ac.be (N. Gillis), punit.sharma@maths.iitd.ac.in (P. Sharma).

[1] N. Gillis acknowledges the support by the Fonds De La Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) under EOS Project no O005318F-RG47, and by the European Research Council (ERC starting grant no 679515).

[2] P. Sharma acknowledges the support of the DST-Inspire Faculty Award (MI01807-G) by Government of India, Institute SEED Grant (NPN5R) and FIRP project (MI02089) by IIT Delhi.

https://doi.org/10.1016/j.laa.2020.02.008

1. Introduction

Consider a continuous linear-time invariant (LTI) system in the form

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t),$$

where, for all $t \in \mathbb{R}$, $x(t) \in \mathbb{R}^n$ is the state, $u(t) \in \mathbb{R}^m$ is the control input, $y(t) \in \mathbb{R}^p$ is the measured output, $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$, and $C \in \mathbb{R}^{p,n}$. Such a system is called stable if all eigenvalues of the matrix A are in the closed left half of the complex plane and those on the imaginary axis are semisimple. Similarly, it is called asymptotically stable if all eigenvalues of A are in the open left half of the complex plane; see for example [4,2]. The notion of stabilizing the system pair (A, B) using feedback controllers is a fundamental one, and is referred to as the static-state feedback (SSF) problem. It requires to find $K \in \mathbb{R}^{m,n}$ such that $A - BK$ is stable. The first goal of this paper is to solve the SSF problem; this can be divided into two parts:

1) Feasibility. Check the existence of a feedback matrix K such that $A - BK$ is stable.
2) Optimization. If the problem is feasible, minimize the norm of the feedback matrix, that is, solve

$$\inf_K \|K\| \quad \text{such that} \quad A - BK \text{ is stable},$$

where $\|\cdot\|$ is a given norm such as the ℓ_2 norm, $\|\cdot\|_2$ or the Frobenius norm, $\|\cdot\|_F$.

The second goal is to consider the analogous problem for system triplets (A, B, C), referred to as the static-output feedback (SOF) problem. The SOF problem requires to find $K \in \mathbb{R}^{m,p}$ such that $A - BKC$ is stable; see [29] for a survey on the SOF problem. This decision problem is believed to be NP-hard as no polynomial-time algorithm is known. Moreover, if extra constraints are imposed on the entries of the static controller, then this decision problem is NP-hard [23,5]. As a consequence, the minimal-norm SOF problem, for which the norm of K is minimized, is also believed to be NP-hard. For more discussion on the hardness of this problem, we refer to the recent paper by Peretz [25] and the references therein. The solution of the SOF problem is important for systems which models structural dynamics, and naturally needs a static feedback that can be built into the structure [27,33,34,26,25]. It was shown that optimal SOFs may achieve similar performance as optimal dynamic feedbacks.

Approaches to tackle the SOF problem include BMI parametrizations [17], SDP programming based approaches [22], and LMI based parametrizations [18,19,6]. A characterization of SOFs of a given LTI system triplet, in terms of some constrained Riccati equation is given by Peretz in [24], and he also proposed a randomized algorithm to minimize the norm of the SOF [25].

Recently, in [12], a parametrization of the set of all stable matrices was obtained in terms of dissipative Hamiltonian (DH) systems. A matrix $A \in \mathbb{R}^{n,n}$ is called a *DH matrix* if $A = (J - R)Q$ for some $J, R, Q \in \mathbb{R}^{n,n}$ such that $J^T = -J$, R is positive semidefinite and Q is positive definite. A matrix A is stable if and only if it is a DH matrix. This parametrization has been used to solve several nearness problems for LTI systems [21, 11,13,10]. Motivated by this, we give a new parametrization to the set of all SSFs and SOFs in terms of DH matrices.

1.1. Contribution and outline of the paper

In this paper, we provide a complete characterization of the SSFs for a system pair (A, B) in terms of matrix triplets (J, R, Q), where J is skew-symmetric, R is positive semidefinite, and Q is positive definite. This is a new alternative way to parametrize the set of SSFs. This leads to a simple computational method to check the feasibility of the SSF problem by solving a convex semidefinite program (SDP). This parametrization results in a reformulation of the nonconvex optimization problem of minimizing the norm of the SSF into an equivalent optimization problem where the feasible set is convex for which we propose a sequential SDP (SSDP) method. We then extend this approach to provide a characterization for the SOF problem. Again, this parametrization provides a computational method to check the existence of an SOF by solving an optimization problem where the feasible set is convex, for which we propose an SSDP method. Although we cannot guarantee to obtain a feasible solution in all cases due to the complexity of the SOF problem, we are able in many cases to get better solutions than competitive methods (that is, provide a feasible solution with smaller norm). Finally, it has to be noted that our new characterization is very flexible. First, it can be used to minimize any convex function of the feedback matrix K. Second, it can be easily extended to require the eigenvalues of the stabilized matrix $A - BKC$ to belong to other sets than the left half of the complex plane, namely, to conic sectors, vertical strips and disks [7]. This is achieved by adding specific convex constraints to the matrix triplets (J, R, Q); see the discussion in Section 5.4. For example, our characterization could directly be used to compute SSFs and SOFs for discrete-time systems (where the eigenvalues of $A - BKC$ must belong to the unit disk).

The paper is organized as follows. In Section 2, we state some preliminary results from the literature. In Section 3, we give a complete characterization of the static-state stabilizing feedbacks for a system pair (A, B) in terms of DH matrices. In Section 4, we extend these results for the SOF problem of a system triplet (A, B, C). In Section 5, using our characterizations based on DH matrices, we propose two algorithms to minimize the norm of stabilizing feedback matrices: one for the SSF problem (Algorithm 1) and one for the SOF problem (Algorithm 2). In Section 6, we illustrate the effectiveness of Algorithm 2 compared to the two state-of-the-art methods for the SOF problem proposed in [3,25] on numerous numerical examples from the COMPLeIB library [20]. Numerical results for Algorithm 1 on the SSF problem are reported in Appendix A.

Notation Throughout the paper, X^T and $X^\dagger$ stand for the transpose and Moore-Penrose pseudoinverse of a real matrix X, respectively. We write $X \succ 0$ and $X \succeq 0$ ($X \preceq 0$) if X is symmetric and positive definite or positive semidefinite (symmetric negative semidefinite), respectively. By I_m we denote the identity matrix of size $m \times m$, and by null(X) we denote the null space of X. The eigenvalue spectrum of a matrix X is denoted by $\Lambda(X)$. For a given matrix triplet (A, B, C), where $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{n,m}$ and $C \in \mathbb{R}^{p,n}$, we define

$$\mathcal{K}(A, B, C) := \{K \in \mathbb{R}^{m,p} \mid A - BKC \text{ is stable}\}, \tag{1.1}$$

and

$$\mathcal{W}(A, B, C) := \left\{W \in \mathbb{R}^{n,n} \mid A - BB^\dagger WC^\dagger C \text{ is stable}\right\}. \tag{1.2}$$

For a matrix pair (A, B), we use the notation $\mathcal{K}_R(A, B)$ for $\mathcal{K}(A, B, I_p)$ and $\mathcal{W}_R(A, B)$ for $\mathcal{W}(A, B, I_p)$. For the matrix triplet (A, I_m, C), we use the notation $\mathcal{K}_L(A, C)$ for $\mathcal{K}(A, I_m, C)$ and $\mathcal{W}_L(A, C)$ for $\mathcal{W}(A, I_m, C)$.

2. Preliminaries

In this section, we present results from the literature that will be useful in the following sections.

Let us first recall the definition of a DH matrix: A matrix $A \in \mathbb{R}^{n,n}$ is said to be a DH matrix if $A = (J - R)Q$ for some $J, R, Q \in \mathbb{R}^{n,n}$ such that $J^T = -J$, $R \succeq 0$ and $Q \succ 0$. The set of stable matrices is characterized as the set of DH matrices in the following.

Theorem 1. [12, Lemma 2] *Let $A \in \mathbb{R}^{n,n}$. Then A is stable if and only if A is a DH matrix.*

For DH matrices, we can easily derive the following lemma for which we provide the proof which will be useful later on.

Lemma 1. [11, Lemma 3] *DH matrices are invariant under orthogonal transformations.*

Proof. Let A be a DH matrix, that is, $A = (J - R)Q$ for some $J^T = -J$, $R \succeq 0$, and $Q \succ 0$. Let U be orthogonal, that is, $U^T U = I_n = UU^T$. Then $U^T AU = U^T (J-R)QU = (U^T JU - U^T RU)U^T QU$ is a DH matrix since $(U^T JU)^T = -U^T JU$, $U^T RU \succeq 0$, and $U^T QU \succ 0$. □

The following lemma will be used to parametrize the set of all feedbacks that stabilize a system pair (A, B) in Section 3 and system triplet (A, B, C) in Section 4.

Lemma 2. [28, Lemma 1.3] *Let $A \in \mathbb{C}^{p,m}$, $B \in \mathbb{C}^{n,q}$, $C \in \mathbb{C}^{p,q}$, and*

$$\Upsilon = \left\{E \in \mathbb{C}^{m,n} \,\middle|\, AEB = C\right\}.$$

Then $\Upsilon \neq \emptyset$ if and only if A, B, C satisfy $AA^\dagger CB^\dagger B = C$. If the latter condition is satisfied then

$$\Upsilon = \left\{A^\dagger CB^\dagger + Z - A^\dagger AZBB^\dagger \,\middle|\, Z \in \mathbb{C}^{n,n}\right\},$$

and

$$\min_{E\in\Upsilon} \|E\|_F = \|A^\dagger CB^\dagger\|_F, \quad \min_{E\in\Upsilon} \|E\|_2 = \|A^\dagger CB^\dagger\|_2.$$

The following well-known lemma gives an equivalent characterization for a positive semidefinite matrix; it will be used in establishing new conditions for the existence of stabilizing feedbacks.

Lemma 3. [1] *Let the integer s be such that $0 < s < n$, and $R = R^T \in \mathbb{R}^{n,n}$ be partitioned as $R = \begin{bmatrix} B & C^T \\ C & D \end{bmatrix}$ with $B \in \mathbb{R}^{s,s}$, $C \in \mathbb{R}^{n-s,s}$ and $D \in \mathbb{R}^{n-s,n-s}$. Then $R \succeq 0$ if and only if*

$$i)\ D \succeq 0, \quad ii)\ \mathrm{null}(D) \subseteq \mathrm{null}(C^T), \quad \text{and} \quad iii)\ B - C^T D^\dagger C \succeq 0.$$

3. DH characterization of static-state stabilizing feedbacks

Let us denote the set of triplets (J, R, Q) that form a DH matrix as follows

$$\mathbb{DH}^n_\succ := \left\{(J, R, Q) \in (\mathbb{R}^{n,n})^3 \mid J^T = -J,\ R \succeq 0,\ Q \succ 0\right\}. \tag{3.1}$$

For a triplet $(J, R, Q) \in (\mathbb{R}^{n,n})^3$, let us also define

$$g(J, R, Q) := B^\dagger(A - (J - R)Q). \tag{3.2}$$

Using Lemma 2, we have the following characterization of the set $\mathcal{K}_R(A, B)$ in terms of triplets $(J, R, Q) \in \mathbb{DH}^n_\succ$.

Theorem 2. *Let $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,m}$. Then,*

$$\begin{aligned}\mathcal{K}_R(A, B) = \Big\{ g(J, R, Q) - (I_m - B^\dagger B)Y \Big| &(J, R, Q) \in \mathbb{DH}^n_\succ, \\ &(I_n - BB^\dagger)(A - (J - R)Q) = 0, Y \in \mathbb{R}^{m,n}\Big\}.\end{aligned} \tag{3.3}$$

Proof. Let us first show that $\mathcal{K}_R(A,B) \neq \emptyset$ if and only if there exists $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ such that $(I_n - BB^\dagger)(A-(J-R)Q) = 0$. Let $K \in \mathcal{K}_R(A,B)$, that is, $A - BK$ is stable. Then by Theorem 1, there exists $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ such that $A - BK = (J-R)Q$. This implies that $BK = A - (J-R)Q$ and thus

$$BB^\dagger(A-(J-R)Q) = BB^\dagger BK = BK = A - (J-R)Q,$$

since $BB^\dagger B = B$. Conversely, let $(I_n - BB^\dagger)(A-(J-R)Q) = 0$, and consider $K = B^\dagger(A-(J-R)Q)$. Then K satisfies

$$A - BK = A - BB^\dagger(A-(J-R)Q) = A - A + (J-R)Q = (J-R)Q.$$

This implies that $A - BK$ is a DH matrix and thus stable, hence $K \in \mathcal{K}_R(A,B)$.

To show (3.3), let K be a matrix of the form $g(J,R,Q) - (I_m - B^\dagger B)Y$ satisfying $(I_n - BB^\dagger)(A-(J-R)Q) = 0$. We have that $A - BK = (J-R)Q$, and thus $K \in \mathcal{K}_R(A,B)$ because $(J,R,Q) \in \mathbb{DH}^n_{\succ}$. This proves the inclusion "$\supseteq$". For the inclusion "$\subseteq$", let $K \in \mathcal{K}_R(A,B)$. By Theorem 1 there exists $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ such that $BK = A - (J-R)Q$. In view of Lemma 2 there exists $Y \in \mathbb{R}^{m,n}$ such that $K = g(J,R,Q) - (I_m - B^\dagger B)Y$ and $(I_n - BB^\dagger)(A-(J-R)Q) = 0$. This proves "$\subseteq$". □

Corollary 1. *For a fixed triplet $(J,R,Q) \in \mathbb{DH}^n_{\succ}$, define*

$$\widetilde{\mathcal{K}}_R(J,R,Q) := \left\{K \in \mathcal{K}_R(A,B) \,\middle|\, K = g(J,R,Q) - (I_m - B^\dagger B)Y,\, Y \in \mathbb{R}^{m,n}\right\}. \quad (3.4)$$

If $\widetilde{\mathcal{K}}_R(J,R,Q) \neq \emptyset$, then

$$\min_{K \in \widetilde{\mathcal{K}}_R(J,R,Q)} \|K\|_F = \|g(J,R,Q)\|_F, \quad \text{and} \quad \min_{K \in \widetilde{\mathcal{K}}_R(J,R,Q)} \|K\|_2 = \|g(J,R,Q)\|_2. \quad (3.5)$$

Proof. This follows immediately from Lemma 2 and Theorem 2. □

The following theorem gives an alternative way to check the existence of a stabilizing feedback. It can be useful in numerical methods as it reduces the number of variables depending on the rank of B; see also Remark 1.

Theorem 3. *Let $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,m}$. Let U be an orthogonal matrix such that $U^T BB^\dagger U = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$, where $\mathrm{rank}(B) = \mathrm{rank}(BB^\dagger) = k$. Let $\hat{A} = U^T A U = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}$, where $\hat{A}_{11} \in \mathbb{R}^{k,k}$ and $\hat{A}_{22} \in \mathbb{R}^{n-k,n-k}$. Then $\mathcal{K}_R(A,B) \neq \emptyset$ if and only if there exist $\hat{Q}(\succ 0) \in \mathbb{R}^{n,n}$, $\hat{J}_{22}, \hat{R}_{22} \in \mathbb{R}^{n-k,n-k}$ and $\hat{J}_{21}, \hat{R}_{21} \in \mathbb{R}^{n-k,k}$ such that $\hat{J}_{22}^T = -\hat{J}_{22}$, $\hat{R}_{22} \succeq 0$, $\mathrm{null}(\hat{R}_{22}) \subseteq \mathrm{null}(\hat{R}_{21}^T)$, and*

$$\begin{bmatrix} \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} = \begin{bmatrix} \hat{J}_{21} - \hat{R}_{21} & \hat{J}_{22} - \hat{R}_{22} \end{bmatrix} \hat{Q}. \quad (3.6)$$

Proof. First suppose that $\mathcal{K}_R(A,B) \neq \emptyset$, and let $K \in \mathcal{K}_R(A,B)$. Then from Theorem 2 there exists $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ such that

$$(I_n - BB^\dagger)(A - (J-R)Q) = 0. \tag{3.7}$$

Multiplying (3.7) by U^T from the left and by U from the right, by using the fact that $U^TU = I_n$, and by setting

$$\hat{J} = U^TJU = \begin{bmatrix} \hat{J}_{11} & -\hat{J}_{21}^T \\ \hat{J}_{21} & \hat{J}_{22} \end{bmatrix}, \hat{R} = U^TRU = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{21}^T \\ \hat{R}_{21} & \hat{R}_{22} \end{bmatrix}, \hat{Q} = U^TQU = \begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{21}^T \\ \hat{Q}_{21} & \hat{Q}_{22} \end{bmatrix},$$

we have that

$$\begin{bmatrix} 0 & 0 \\ 0 & I_{n-k} \end{bmatrix} \left(\begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} - \begin{bmatrix} \hat{J}_{11} - \hat{R}_{11} & -\hat{J}_{21}^T - \hat{R}_{21}^T \\ \hat{J}_{21} - \hat{R}_{21} & \hat{J}_{22} - \mathbb{R}_{22} \end{bmatrix} \begin{bmatrix} \hat{Q}_{11} & \hat{Q}_{21}^T \\ \hat{Q}_{21} & \hat{Q}_{22} \end{bmatrix} \right) = 0.$$

This implies that

$$\begin{bmatrix} \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} = \begin{bmatrix} \hat{J}_{21} - \hat{R}_{21} & \hat{J}_{22} - \hat{R}_{22} \end{bmatrix} \hat{Q}.$$

Note that $\hat{Q} \succ 0$ and $\hat{J}_{22}^T = -\hat{J}_{22}$, because U is orthogonal and $(J,R,Q) \in \mathbb{DH}^n_{\succ}$. Also from Lemma 3 $\hat{R}_{22} \succeq 0$ and $\text{null}(\hat{R}_{22}) \subseteq \text{null}(\hat{R}_{21}^T)$, since $\hat{R} \succeq 0$. This proves the "if" part.

Conversely, suppose that (3.6) holds. Let $\hat{R}_{11}, \hat{J}_{11} \in \mathbb{R}^{k,k}$ be chosen such that $\hat{J}_{11}^T = -\hat{J}_{11}$ and $\hat{R}_{11} - \hat{R}_{21}\hat{R}_{22}^\dagger\hat{R}_{21} \succeq 0$ (for example: $\hat{R}_{11} = \hat{R}_{21}\hat{R}_{22}^\dagger\hat{R}_{21}$) and let $\hat{J} = U^TJU = \begin{bmatrix} \hat{J}_{11} & -\hat{J}_{21}^T \\ \hat{J}_{21} & \hat{J}_{22} \end{bmatrix}$ and $\hat{R} = U^TRU = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{21}^T \\ \hat{R}_{21} & \hat{R}_{22} \end{bmatrix}$. Observe that $\hat{J}^T = -\hat{J}$, $\hat{R} \succeq 0$ (Lemma 3) and

$$\begin{bmatrix} 0 & 0 \\ 0 & I_{n-k} \end{bmatrix} (\hat{A} - (\hat{J} - \hat{R})\hat{Q}) = 0. \tag{3.8}$$

Multiplying (3.8) by U from the left and by U^T from the right, and by using the fact that $U^TU = I_n$, we get

$$(I_n - BB^\dagger)(A - (U\hat{J}U^T - U\hat{R}U^T)U\hat{Q}U^T) = 0.$$

Thus from Theorem 2, $K = B^\dagger(A - (U\hat{J}U^T - U\hat{R}U^T)U\hat{Q}U^T)$ satisfies $A - BK = (U\hat{J}U^T - U\hat{R}U^T)U\hat{Q}U^T$. This implies from Lemma 1 that $A - BK$ is a DH matrix and thus $K \in \mathcal{K}_R(A,B)$. □

It is well known that if the system pair (A,B) is controllable, then $\mathcal{K}_R(A,B) \neq \emptyset$; see [31]. In the following, we obtain a different sufficient condition for the existence of a stabilizing feedback for pair (A,B) which is a corollary of Theorem 3.

Corollary 2. *Let $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,m}$. Let U be an orthogonal matrix such that $U^T BB^\dagger U = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$, where $\operatorname{rank}(B) = \operatorname{rank}(BB^\dagger) = k$. Let $\hat{A} = U^T AU = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}$, where $\hat{A}_{11} \in \mathbb{R}^{k,k}$ and $\hat{A}_{22} \in \mathbb{R}^{n-k,n-k}$. If $\hat{A}_{22}$ is stable, then $\mathcal{K}_R(A, B) \neq \emptyset$.*

Proof. Since $\hat{A}_{22}$ is stable, from Theorem 1 there exist $(\hat{J}_{22}, \hat{R}_{22}, \hat{Q}_{22}) \in \mathbb{DH}_{\succ}^{n-k}$ such that $A_{22} = (\hat{J}_{22} - \hat{R}_{22})\hat{Q}_{22}$. Define

$$\hat{J} = \begin{bmatrix} \hat{J}_{11} & -\hat{J}_{21}^T \\ \hat{J}_{21} & \hat{J}_{22} \end{bmatrix}, \hat{R} = \begin{bmatrix} 0 & 0 \\ 0 & \hat{R}_{22} \end{bmatrix}, \hat{Q} = \begin{bmatrix} I_k & 0 \\ 0 & \hat{Q}_{22} \end{bmatrix},$$

where $\hat{J}_{11} \in \mathbb{R}^{k,k}$ is any skew-symmetric matrix and $\hat{J}_{21} = \hat{A}_{21}$. Note that $(\hat{J}, \hat{R}, \hat{Q}) \in \mathbb{DH}_{\succ}^{n}$. Further define

$$J = U\hat{J}U^T, \quad R = U\hat{R}U^T, \quad \text{and} \quad Q = U\hat{Q}U^T.$$

Proceeding as in Theorem 3, we have that $(I_n - BB^\dagger)(A - (J - R)Q) = 0$, and the matrix $K = B^\dagger(A - (J - R)Q)$ stabilizes the system pair (A, B) hence $K \in \mathcal{K}_R(A, B)$. $\square$

Note that the converse of the above corollary does not hold. For example, consider

$$A = \begin{bmatrix} -0.3633 & -0.2867 & -0.7294 & -2.2033 \\ -1.0206 & -0.1973 & 1.1473 & -0.5712 \\ -3.0730 & 0.4056 & 0.5979 & 0.2140 \\ 0.6263 & -1.4193 & -1.2813 & 0.9424 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0.0937 & -0.9610 \\ -1.1223 & -0.6537 \\ 0.3062 & -1.2294 \\ -1.1723 & -0.2710 \end{bmatrix},$$

where $\Lambda(A) = \{1.9604 + 1.8099i, 1.9604 + 1.8099i, -1.8378, -1.1032\}$. We have

$$B^\dagger = \begin{bmatrix} 0.1089 & -0.3490 \\ -0.3787 & -0.1472 \\ 0.2103 & -0.4607 \\ -0.4269 & -0.0073 \end{bmatrix}^T \text{ and } U = \begin{bmatrix} -0.5879 & 0.0000 & 0.7931 & 0.1595 \\ -0.1803 & -0.6991 & 0.0055 & -0.6919 \\ -0.7866 & 0.1098 & -0.6010 & 0.0893 \\ 0.0562 & -0.7066 & -0.0988 & 0.6985 \end{bmatrix}.$$

For the pair (A, B), the SSF problem is feasible, that is, $\mathcal{K}_R(A, B) \neq \emptyset$ as

$$K = \begin{bmatrix} 3.4237 & 27.5800 & 1.9374 & -35.6683 \\ 10.1752 & -23.1448 & -11.0932 & 20.1984 \end{bmatrix}$$

with $\Lambda(A - BK) = \{-9.9703, -8.7173, -2.4493, -3.1767\}$. However, for the transformed matrix

$$\hat{A} = U^T A U = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}, \quad \text{where} \quad \hat{A}_{22} = \begin{bmatrix} 1.8596 & -0.9818 \\ 1.6202 & 1.0747 \end{bmatrix}$$

we have $\Lambda(A_{22}) = \{1.4672 + 1.1986i, 1.4672 - 1.1986i\}$. This implies $\mathcal{K}_R(A, B) \neq \emptyset$ while $\hat{A}_{22}$ is not stable.

In the following theorem, we summarize the various necessary and sufficient conditions for the existence of a static stabilizing feedback for a given system pair (A, B).

Theorem 4. *Let $A \in \mathbb{R}^{n,n}$ and $B \in \mathbb{R}^{n,m}$. Then the following are equivalent.*

1) $\mathcal{W}_R(A, B) \neq \emptyset$.
2) $\mathcal{K}_R(A, B) \neq \emptyset$.
3) *There exists $(J, R, Q) \in \mathbb{DH}^n_{\succ}$ such that $A - BK = (J - R)Q$ for some $K \in \mathbb{R}^{m,n}$.*
4) *There exists $(J, R, Q) \in \mathbb{DH}^n_{\succ}$ such that $A - BB^\dagger W = (J - R)Q$ for some $W \in \mathbb{R}^{n,n}$.*
5) *There exists $(J, R, Q) \in \mathbb{DH}^n_{\succ}$ such that $(I_n - BB^\dagger)(A - (J - R)Q) = 0$.*
6) *There exist $\hat{Q}(\succ 0) \in \mathbb{R}^{n,n}$, $\hat{J}_{22}, \hat{R}_{22} \in \mathbb{R}^{n-k,n-k}$ and $\hat{J}_{21}, \hat{R}_{22} \in \mathbb{R}^{n-k,k}$ such that $\hat{J}_{22}^T = -\hat{J}_{22}$, $\hat{R}_{22} \succeq 0$, $\mathrm{null}(\hat{R}_{22}) \subseteq \mathrm{null}(\hat{R}_{21}^T)$, and*

$$\begin{bmatrix} \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} = \begin{bmatrix} \hat{J}_{21} - \hat{R}_{21} & \hat{J}_{22} - \hat{R}_{22} \end{bmatrix} \hat{Q},$$

where $\hat{A} = U^T A U = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}$ with $\hat{A}_{11} \in \mathbb{R}^{k,k}$, $\hat{A}_{22} \in \mathbb{R}^{n-k,n-k}$ and U is an orthogonal matrix such that $U^T BB^\dagger U = \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathrm{rank}(B) = k$.

Proof. 1) $\Leftrightarrow$ 2): This follows immediately from the relationship between the two sets $\mathcal{W}_R(A, B)$ and $\mathcal{K}_R(A, B)$. In fact, for any $W \in \mathcal{W}_R(A, B)$ and $Y \in \mathbb{R}^{m,n}$, we have $K = B^\dagger W + (I_m - B^\dagger B)Y \in K_R(A, B)$, while for any $K \in \mathcal{K}_R(A, B)$ and $N \in \mathbb{R}^{n,n}$, we have $W = BK + (I_n - BB^\dagger)N \in \mathcal{W}_R(A, B)$ [24].

3) $\Rightarrow$ 2) and 4) $\Rightarrow$ 1) follow trivially, and 2) $\Rightarrow$ 3) and 1) $\Rightarrow$ 4) follow from Theorem 1.

2) $\Leftrightarrow$ 5): This follows from Theorem 2.

2) $\Leftrightarrow$ 6): This follows from Theorem 3. □

4. DH characterization of static-output stabilizing feedbacks

We first state a result similar to Theorem 2 that characterizes the set of SOFs in terms of DH matrices. For this, let $(J, R, Q) \in (\mathbb{R}^{n,n})^3$ and define

$$f(J, R, Q) := B^\dagger (A - (J - R)Q) C^\dagger.$$

Theorem 5. *Let (A, B, C) be a system triplet. Then*

$$K(A, B, C) = \Big\{ f(J, R, Q) + Z - B^\dagger BZCC^\dagger \Big| (J, R, Q) \in \mathbb{DH}^n_{\succ}, Z \in \mathbb{R}^{m \times p} \text{ and}$$

$$Bf(J,R,Q)C = A-(J-R)Q\Big\}.$$

Proof. The proof is similar to Theorem 2. $\square$

A corollary similar to Corollary 1 is given in the following.

Corollary 3. *For a fixed triplet $(J,R,Q) \in \mathbb{DH}^n_{\succ}$, define*

$$\widetilde{\mathcal{K}}(J,R,Q) := \left\{K \in \mathcal{K}(A,B,C) \middle|\ K = f(J,R,Q) + Z - B^\dagger BZCC^\dagger,\ Z \in \mathbb{R}^{m,p}\right\}. \quad (4.1)$$

If $\widetilde{\mathcal{K}}(J,R,Q) \neq \emptyset$, then

$$\min_{K\in\widetilde{\mathcal{K}}(J,R,Q)} \|K\|_F = \|f(J,R,Q)\|_F, \quad \textit{and} \quad \min_{K\in\widetilde{\mathcal{K}}(J,R,Q)} \|K\|_2 = \|f(J,R,Q)\|_2. \quad (4.2)$$

It is easy to see that if the matrix triplet (A,B,C) is stabilizable, then the pairs (A,B) and (A,C) are necessarily stabilizable. Indeed, if $K \in \mathcal{K}(A,B,C)$, then $KC \in \mathcal{K}_R(A,B)$ and $BK \in \mathcal{K}_L(A,C)$. In general, however, stabilization of (A,B) and (A,C) do not guarantee the stabilization of (A,B,C). To ensure this, there must exist $Y \in \mathcal{W}_R(A,B)$ and a $Z \in \mathcal{W}_L(A,C)$ such that $BB^\dagger Y = ZC^\dagger C$ [25, Lemma 3.1]. In the following, we give a different sufficient condition for the stabilizability of (A,B,C) in terms of DH matrices; if the stabilizability of (A,B) and (A,C) is determined simultaneously by the same stable matrix, that is, if there exists a $Y \in \mathcal{K}_R(A,B)$ and a $Z \in \mathcal{K}_L(A,C)$ such that $A - BY = A - ZC$, then (A,B,C) is stabilizable.

Theorem 6. *Let (A,B,C) be a given system triplet. Then (A,B,C) is stabilizable if and only if there exists a $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ such that*

$$(I_n - BB^\dagger)(A-(J-R)Q) = 0 \quad \textit{and} \quad (A-(J-R)Q)(C^\dagger C - I_n) = 0. \quad (4.3)$$

If the later conditions are satisfied, then for all SOFs related to such (J,R,Q) we have

$$\widetilde{\mathcal{K}}(J,R,Q) = \left\{f(J,R,Q) + (I_m - B^\dagger B)Y + Z(I_s - CC^\dagger)\ \middle|\ Y,Z \in \mathbb{R}^{m,s}\right\}. \quad (4.4)$$

Proof. It is easy to check that for a given $(J,R,Q) \in \mathbb{DH}^n_{\succ}$ satisfying (4.3), the matrix $f(J,R,Q) \in \mathcal{K}(A,B,C)$. Conversely, let $K \in \mathcal{K}(A,B,C)$, that is, $A-BKC$ is stable. This implies $A - BKC = (J-R)Q$ for some $(J,R,Q) \in \mathbb{DH}^n_{\succ}$, and thus $KC \in \mathcal{K}_R(A,B)$ and $BK \in \mathcal{K}_L(A,C)$. Hence as an application of Lemma 2, (J,R,Q) satisfies (4.3). Further, (4.4) follows immediately by using the fact that $BB^\dagger B = B$ and $CC^\dagger C = C$. $\square$

5. Computing stabilizing feedback matrices

In this section, we exploit the results obtained in the previous sections and present a new framework based on DH matrices to attack the SSF and SOF problems.

5.1. Stabilization of a system pair (A, B)

In this section, we focus on stabilizing a system pair (A, B), that is, finding K such that $A - BK$ is stable. We propose Algorithm 1 which consists in two mains steps described in the next subsections: first finding a feasible solution and then improving this solution.

5.1.1. Feasibility problem

A necessary and sufficient condition was obtained in Theorem 2 for the feasibility of the static feedback problem in terms of DH matrices: $\mathcal{K}_R(A, B) \neq \emptyset$ if and only if there exists $(J, R, Q) \in \mathbb{DH}^n_{\succ}$ such that $(I_n - BB^\dagger)(A - (J - R)Q) = 0$. Trying to find a feasible solution of the latter equation can be done by considering the following optimization problem

$$\mu := \inf_{J,R,Q \in \mathbb{R}^{n,n}, J^T = -J, R \succeq 0, Q \succ 0} \left\|(I_n - BB^\dagger)(A - (J - R)Q)\right\|, \tag{5.1}$$

and checking whether $\mu = 0$, that is, $\mathcal{K}_R(A, B) \neq \emptyset$ if and only if $\mu = 0$. Note that the optimization problem in (5.1) is nonconvex due to the presence of the nonlinear term $(J - R)Q$ in the objective, however, it can be transformed into an equivalent convex optimization problem by means of defining a new variable $P = Q^{-1}$ as follows. Since Q is positive definite,

$$(I_n - BB^\dagger)(A - (J - R)Q) = 0 \quad \Longleftrightarrow \quad (I_n - BB^\dagger)(AQ^{-1} - J + R) = 0.$$

Defining,

$$\tilde{\mu} = \inf_{J,R,P \in \mathbb{R}^{n,n}, J^T = -J, R \succeq 0, P \succ 0} \left\|(I_n - BB^\dagger)(AP - J + R)\right\|, \tag{5.2}$$

we have that $\mu = 0$ if and only if $\tilde{\mu} = 0$, and the optimization problem (5.2) is convex. There is a scaling degree of freedom between (J, R) and $P = Q^{-1}$ since $(\alpha J, \alpha R)$ and αP for any $\alpha > 0$ leads to an equivalent solution as $(I_n - BB^\dagger)(AP - J + R) = 0 \iff (I_n - BB^\dagger)(\alpha AP - \alpha J + \alpha R) = 0$. To avoid this degree of freedom, one may choose for example α such that $\|J - R\|_2 = \|P^{-1}\|_2$ as done in [12]. In our case, we use this degree of freedom by assuming w.l.o.g. that $P \succeq I_n$. Moreover, the feasible set in (5.2) is neither open (due to constraint $R \succeq 0$) nor closed (due to constraint $P \succ 0$) and replacing the feasible set in (5.2) by its closure does not change the value of the infimum in (5.2). Therefore

$$\tilde{\mu} = \inf_{J,R,Q \in \mathbb{R}^{n,n}, J^T = -J, R \succeq 0, P \succeq I_n} \left\|(I_n - BB^\dagger)(AP - J + R)\right\|. \tag{5.3}$$

Thus checking feasibility is equivalent to check that the value of $\tilde{\mu}$ in (5.3) is zero or not. The problem (5.3) is convex. More precisely, it is a semidefinite program (SDP) which can be solved efficiently with dedicated solvers.

Remark 1. Note that one can use Theorem 3 to obtain an equivalent optimization problem with less variables (depending on the rank of B). Defining

$$\nu := \inf_{J_{22},R_{22}\in\mathbb{R}^{n-k,n-k},J_{21}\in\mathbb{R}^{n-k,k},Q\in\mathbb{R}^{n,n}} \left\| \begin{bmatrix} \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} - \begin{bmatrix} J_{21}-R_{21} & J_{22}-R_{22} \end{bmatrix} Q \right\|$$
$$\text{such that} \quad Q \succ 0,\ J_{22}^T = -J_{22},\ R_{22} \succeq 0,\ \text{null}(R_{22}) \subseteq \text{null}(R_{21}^T),$$

we have $\mathcal{K}_R(A,B) \neq \emptyset$ if and only if $\nu = 0$. Note that if $R_{22} \succ 0$, then the condition $\text{null}(R_{22}) \subseteq \text{null}(R_{21}^T)$ is always met. Therefore dropping this condition does not make any difference in our algorithm as the set of positive definite matrices is dense in the set of positive semidefinite matrices. Therefore an equivalent reformulation of (5.3) is given by

$$\nu = \inf_{J_{22},R_{22}\in\mathbb{R}^{n-k,n-k},J_{21}\in\mathbb{R}^{n-k,k},P\in\mathbb{R}^{n,n}} \left\| \begin{bmatrix} \hat{A}_{21} & \hat{A}_{22} \end{bmatrix} P - \begin{bmatrix} J_{21}-R_{21} & J_{22}-R_{22} \end{bmatrix} \right\|$$
$$\text{such that} \quad P \succeq I_n,\ J_{22}^T = -J_{22},\ R_{22} \succeq 0.$$

5.1.2. Optimization problem

Suppose that the SSF problem is feasible, that is, $\mathcal{K}_R(A,B) \neq \emptyset$, and we want to solve the minimization problem

$$\inf_{K\in\mathcal{K}_R(A,B)} \|K\|. \tag{5.4}$$

In view of Theorem 2 and Corollary 1, we can equivalently write (5.4) as

$$\inf_{(J,R,Q)\in\mathbb{DH}_{\succ}^n} \ \inf_{K\in\widetilde{\mathcal{K}}_R(J,R,Q)} \|K\|$$
$$= \inf_{(J,R,Q)\in\mathbb{DH}_{\succ}^n,\ (I_n-BB^\dagger)(A-(J-R)Q)=0} \|B^\dagger(A-(J-R)Q)\|. \tag{5.5}$$

In view of the formulation (5.5), a simple algorithm that can be used is a block coordinate descent (BCD) method: optimize alternatively over variables (J,R) for Q fixed, and Q for (J,R) fixed. In fact, the subproblems are SDPs hence can be solved efficiently. However, we have observed in practice that BCD seems to get stuck at saddle points; see Appendix A for numerical results. For this reason, we have developed another algorithm that is based on sequential semidefinite programming (SSDP). As before, let us denote $P = Q^{-1}$, and reformulate (5.5) as

$$\inf_{J,R,P\in\mathbb{R}^{n,n},J^T=-J,R\succeq 0,P\succ 0,(I_n-BB^\dagger)(AP-(J-R))=0} \left\|B^\dagger(A-(J-R)P^{-1})\right\|. \tag{5.6}$$

Note that the feasible set is convex, with linear matrix inequalities (LMIs) and linear constraints. Given an initial solution (J,R,P), we look for $(\Delta J,\Delta R,\Delta P)$ such that

Algorithm 1 Stabilizing a system pair (A, B).

Input: The n-by-n matrix A, the n-by-m matrix B, choice of a norm, step-size bound $0 < \underline{\epsilon} \ll 1$.
Output: An approximate solution K to $\min_K \|K\|$ such that $A - BK$ is stable.

% Initialization phase
Initialize $\epsilon = 1$.
Initialize (J, R, P) as the optimal solution to (5.3).
% Optimization phase
Define $F(J, R, P) = \left\|B^\dagger(A - (J - R)P^{-1})\right\|$.
while some stopping criterion is met, or a maximum number of iterations is reached **do**
  Solve (5.7) to obtain $(\Delta J, \Delta R, \Delta P)$.
  while $F(J + \Delta J, R + \Delta R, P + \Delta P) \geq F(J, R, P)$ and $\epsilon > \underline{\epsilon}$ **do**
    Reduce ϵ.
    Solve (5.7) to obtain $(\Delta J, \Delta R, \Delta P)$.
  end while
  Set $(J, R, P) = (J + \Delta J, R + \Delta R, P + \Delta P)$
  Increase ϵ.
end while
Return $K = B^\dagger(A - (J - R)P^{-1})$.

$(J + \Delta J, R + \Delta R, P + \Delta P)$ is a better solution than (J, R, P). To do so, we linearize the term $(A - (J + \Delta J - (R + \Delta R))(P + \Delta P)^{-1})$ by using

$$(P + \Delta P)^{-1} \approx P^{-1} - P^{-1}\Delta P P^{-1},$$

and removing the non-linear terms appearing in the product of the two components, that is, we use the following approximation:

$$\begin{aligned} & A - (J + \Delta J - (R + \Delta R))(P + \Delta P)^{-1} \\ & \quad \approx A - (J + \Delta J - (R + \Delta R))P^{-1} + (J - R)P^{-1}\Delta P P^{-1}. \end{aligned}$$

This results in the following optimization problem

$$\begin{aligned} \inf_{\Delta J, \Delta R, \Delta P \in \mathbb{R}^{n,n}} & \left\|B^\dagger\left(A - (J + \Delta J)P^{-1} + (R + \Delta R)P^{-1} + (J - R)P^{-1}\Delta P P^{-1}\right)\right\| \\ \text{such that} \quad & \Delta J^T = -\Delta J, R + \Delta R \succeq 0, P + \Delta P \succ 0, \\ & (I - BB^\dagger)(A\Delta P - (\Delta J - \Delta R)) = 0, \\ & \|\Delta J\| \leq \epsilon\|J\|, \|\Delta R\| \leq \epsilon\|R\|, \|\Delta P\| \leq \epsilon\|P\|. \end{aligned} \tag{5.7}$$

Similar to a trust-region method, the value of ϵ is updated in the curse of the algorithm. As long as the error of $(J + \Delta J, R + \Delta R, P + \Delta P)$ is larger than that of (J, R, P), ϵ is decreased. For the next step, ϵ is increased to allow a larger trust-region radius.

5.2. Stabilization of a system triplet (A, B, C)

In this section, we focus on stabilizing a system triplet (A, B, C), that is, finding K such that $A - BKC$ is stable. We propose Algorithm 2 which, as for Algorithm 1, consists in two mains steps described in the next subsections.

5.2.1. Feasibility problem

In view of Theorem 6, $\mathcal{K}(A, B, C) \neq \emptyset$ if and only if there exists $(J, R, Q) \in \mathbb{DH}^n_{\succ}$ such that

$$(I_n - BB^\dagger)(A - (J - R)Q) = 0 \quad \text{and} \quad (A - (J - R)Q)(C^\dagger C - I_n) = 0.$$

Finding a feasible solution can be done by considering the following optimization problem

$$\rho := \inf_{J,R,Q \in \mathbb{R}^{n,n}, J^T = -J, R \succeq 0, Q \succ 0} \left\|(I_n - BB^\dagger)(A - (J - R)Q)\right\| + \left\|(A - (J - R)Q)(C^\dagger C - I_n)\right\|, \tag{5.8}$$

since $\mathcal{K}(A, B, C) \neq \emptyset$ if and only if $\rho = 0$. Unlike (5.3), a change of variable as $P = Q^{-1}$ in (5.8) will not result in a convex optimization problem due to the second term in (5.8). This was expected since the stabilization of a matrix triplet is believed to be NP hard; see Section 1. Therefore to solve (5.8), we apply SSDP, as done for (5.6), by solving, at each iteration, the following optimization problem:

$$\begin{aligned} \inf_{\Delta J, \Delta R, \Delta P \in \mathbb{R}^{n,n}} & \left\|(I_n - BB^\dagger)\left(A - (J + \Delta J)P^{-1} + (R + \Delta R)P^{-1} + (J - R)P^{-1}\Delta P P^{-1}\right)\right\| \\ & + \left\|\left(A - (J + \Delta J)P^{-1} + (R + \Delta R)P^{-1}\right.\right. \\ & \left.\left. + (J - R)P^{-1}\Delta P P^{-1}\right)(C^\dagger C - I_n)\right\| \\ \text{such that} \quad & \Delta J^T = -\Delta J, R + \Delta R \succeq 0, P + \Delta P \succ 0, \\ & \|\Delta J\| \leq \epsilon \|\Delta J\|, \|\Delta R\| \leq \epsilon \|\Delta R\|, \|\Delta P\| \leq \epsilon \|\Delta P\|. \end{aligned} \tag{5.9}$$

Algorithm 2 (steps 1-16) summarizes this approach.

5.2.2. Optimization problem

In view of Theorem 5, we want to minimize the norm of feasible feedback K by solving

$$\begin{aligned} \inf_{(J,R,Q) \in \mathbb{DH}^n_{\succ}} \quad & \left\|B^\dagger (A - (J - R)Q)C^\dagger\right\|, \\ \text{such that} \quad & (I_n - BB^\dagger)(A - (J - R)Q) = 0, \text{ and} \\ & (A - (J - R)Q)(C^\dagger C - I_n) = 0. \end{aligned} \tag{5.10}$$

To solve (5.10), we cannot use SSDP because the linear constraints cannot be linearized exactly (we would obtain an infeasible solution after one step). Therefore, in this case, we resort to BCD: alternatively solve (5.10) for (J, R) with Q fixed, and then for Q with (J, R) fixed; see steps 18-22 of Algorithm 2.

Algorithm 2 Stabilizing a system triplet (A, B, C).

Input: The n-by-n matrix A, the n-by-m matrix B, the p-by-n matrix C, choice of a norm, accuracy $0 < \delta \ll 1$, step-size bound $0 < \underline{\epsilon} \ll 1$, an initialization strategy (identity, random, ABI or ACI).
Output: If it succeeds, an approximate solution K to $\min_K \|K\|$ such that $A - BKC$ is stable.

% Initialization phase
Initialize (J, R, P) using the initialization strategy; see Section 5.2.3.
Initialize $\epsilon = 1$.
Define $G(J, R, P) = \|(I_n - BB^\dagger)(A - (J - R)Q)\| + \|(A - (J - R)Q)(C^\dagger C - I_n)\|$.
while $G(J + \Delta J, R + \Delta R, P + \Delta P) \geq \delta$, or a maximum number of iterations is reached **do**
    Solve (5.9) to obtain $(\Delta J, \Delta R, \Delta P)$.
    while $G(J + \Delta J, R + \Delta R, P + \Delta P) \geq G(J, R, P)$ and $\epsilon > \underline{\epsilon}$ **do**
        Reduce ϵ.
        Solve (5.9) to obtain $(\Delta J, \Delta R, \Delta P)$.
    end while
    Set $(J, R, P) = (J + \Delta J, R + \Delta R, P + \Delta P)$
    Increase ϵ.
end while
if $G(J, R, P) \geq \delta$ **then**
    The algorithm failed to find a feasible solution, STOP.
end if
% Optimization phase
$Q = P^{-1}$.
while some stopping criterion is met, or a maximum number of iterations is reached **do**
    Set (J, R) as the optimal solution to (5.10) for Q fixed.
    Set Q as the optimal solution to (5.10) for (J, R) fixed.
end while
Return $K = B^\dagger(A - (J - R)Q)C^\dagger$.

5.2.3. Initialization

Algorithm 2 needs to be initialized with some matrices (J, R, P). We propose four different initializations. In all cases, we choose P and then set (J, R) as an optimal solution of (5.8) with $Q = P^{-1}$.

1) Identity. We simply pick $P = I_n$.
2) Random. We first generate the n-by-n matrix R where each entry is drawn using the normal distribution of mean 0 and standard deviation 1 (`randn(n,n)` in Matlab) and then set $P = (RR^T)^{1/2}$. (We use the square root so that P has a smaller condition number.)
3) ABI. We take P as the optimal solution of (5.3). This means that we choose P such that there exists J and R where $K = B^\dagger(A - (J - R)P^{-1})$ stabilizes the triplet (A, B, I_n) hence the name 'ABI'.
4) AIC. This is similar to ABI except that we choose P such that there exists J and R where $K = (A - (J - R)P^{-1})C^\dagger$ stabilizes the triplet (A, I_n, C) hence the name 'AIC'.

5.3. Computational cost and convergence

Algorithms 1 and 2 require to solve SDPs with matrix variables of dimension $n \times n$. In our code, we use an interior-point method to solve these SDPs (see Section 6 for more details) which requires $\mathcal{O}(n^6)$ operations per iteration (each step requires to solve a linear system with $\mathcal{O}(n^2)$ variables, like in the Newton's method). Hence, on a standard laptop, n can be up to about 50, and our algorithms do not scale well. An important direction of further research is the design of faster algorithms to solve our SDPs such as first-order methods.

In terms of convergence, Algorithm 2 inherits the results for block coordinate descent methods. In particular, since the optimization phase is an exact two-block coordinate descent scheme (that is, there are two blocks of variables that are alternatively optimized exactly), it is guaranteed that every limit point of the sequence generated by Algorithm 2 is a stationary point of the corresponding optimization problem [15]. The convergence to stationary points of limit points of the iterates generated by Algorithm 1 is also guaranteed within the framework of sequential quadratic approximation of the objective function within a trust-region method; see [8] and the references therein. Note that Algorithms 1 and 2 guarantee the objective function to decrease at each iteration, hence the objective function value converges in both cases since it is bounded below (it is nonnegative).

5.4. Extension to Ω-stabilization

Our approach to tackle the SOF and SSF problems can be extended to find feedbacks that make the system Ω-stable. A matrix is said to be Ω-stable if its eigenvalues belong to the set $\Omega \subset \mathbb{C}$. In [7], a parametrization of Ω-stable matrices was obtained in terms of DH matrices that satisfy additional LMI constraints, where Ω is the intersection of specific regions in the complex plane; namely conic sectors, vertical strips, and disks. For example, one may want the real parts of the eigenvalues of $A-BKC$ to be strictly smaller than some given negative value so that $A-BKC$ is robustly stable (see also Remark 2 below). Using these parametrizations, the results obtained in Sections 3 and 4 can be directly extended to obtain a characterization of static stabilizing feedbacks that guarantee Ω-stability in terms of DH matrices. The corresponding optimization problems (5.7) and (5.9) would be subject to additional LMIs on the variables J, R and Q depending on the Ω region; see [7, Theorems 1-3]. For example, our algorithms can be directly adapted to handle discrete-time systems where the eigenvalues of $A-BKC$ are required to belong to the unit disk–this simply requires to use the appropriate semidefinite constraints on the triplet (J, R, Q).

Remark 2. Note that with the current version of the code, replacing A with $A+\rho I$ allows to compute a feedback matrix that makes the real parts of the eigenvalues of $A-BKC$ smaller than $-\rho$. In fact, the real parts of the eigenvalues of $A-BKC$ are smaller than $-\rho$ if and only if the real parts of the eigenvalues of $A+\rho I-BKC$ are smaller than 0.

6. Numerical experiments

In this section, we run our algorithms on the data sets from the library [20]. To solve the convex optimization subproblems involving LMIs, we used the interior point method

SDPT3 (version 4.0) [30,32] with CVX as a modeling system [9,14]. Our code is available from https://sites.google.com/site/nicolasgillis/code and the numerical examples presented below can be directly run from this online code. All tests are preformed using Matlab R2015a on a laptop Intel CORE i7-7500U CPU @2.7 GHz 24Go RAM. It has to be noted that our algorithms are very flexible when it comes to the choice of the norm to be minimized. In fact, CVX can handle many norms. However, we present results for the ℓ_2 norm which is standard in the literature.

We focus in this section on the SOF problem, which is more challenging. Numerical results for the stabilization of matrix pairs (A, B) are reported in Appendix A. We compare Algorithm 2 with the solutions computed by two state-of-the-art algorithms:

- The algorithm proposed in [25] is a randomized approximation algorithm. It works in two phases, similarly as Algorithm 2. In the first phase (RS-PHASE-I, RS stands for Ray-Shooting), the algorithm looks for a feasible solution and, in the second phase (RS-PHASE-II), it improves the solution by minimizing its ℓ_2-norm while remaining feasible. Initially, we tried to reproduce the results in [25] but this is impossible due to the randomized part of the algorithm. Moreover, there are several parameters of the algorithm that need to be fine-tuned to obtain good solutions, and we were not able to produce solutions as good as those presented in [25]. This is the reason why we prefer to report the results from [25]. Unfortunately, the author only provided extensive numerical results for the RS-PHASE-I of his algorithm [25, Table 8], and only 4 solutions obtained after RS-PHASE-II. Hence, we only compare to these solutions in this paper.
- The algorithm proposed in [3] uses the HIFOO non-linear optimization Toolbox; see also [16]. The algorithm, which we will refer to as HIFOO, relies on random initialization so, for the same reason as above, it is not possible to reproduce their results.

For Algorithm 2, we will use the four initializations presented in Section 5.2.3. For random initialization, we report the best result out of 10 initializations. We use the ℓ_2 norm as in [25], and the parameters $\underline{\epsilon} = \delta = 10^{-9}$; while when ϵ is decreased (resp. increased), it is divided (resp. multiplied) by two (steps 9 and 13 of Algorithm 1, and steps 8 and 12 of Algorithm 2). For both the initialization and optimization phases, we limit the number of iterations to 100. The initialization phase is stopped if the error is below 10^{-9}. The optimization phase is stopped if the error is not decreased by at least 10^{-4} between two iterations.

For the stabilization of matrix pairs, we report the solutions found by Algorithm 1 in the Appendix A, and all the results can be rerun from our code available online.

6.1. The four examples from [25, Section 5.1]

Let us first compare our algorithm with RS-PHASE-II on the examples presented in [25, Section 5.1]; see Table 6.1. The solution reported for Algorithm 2 is the best one

Table 6.1
Comparison of the stabilizing feedback matrices K, their norm $\|K\|_2$ (the lowest norm is highlighted in bold), $\max_i \operatorname{Re} \lambda_i(A - BKC)$, and the computational time to obtain them.

	Algorithm 2	RS-PHASE-II
AC7	K = [-0.2638 -0.2506] $\|K\|_2 = \mathbf{0.36}$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0004$ 7.4 s.	K = [-0.5963 -0.0669] $\|K\|_2 = 0.60$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0056$ 0.17 s.
AC8	K = [-0.0112 0.0049 0.0129 -0.0008 -0.0128] $\|K\|_2 = \mathbf{0.02}$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0257$ 7.5 s.	K = [-2.0783 0.2537 0.9202 -0.0382 -0.8338] $\|K\|_2 = 2.43$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.186$ 0.03 s.
HE1	K = [0.1221; -0.3974] $\|K\|_2 = 0.42$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0435$ 61 s.	K = [-0.0691; -0.4227] $\|K\|_2 = \mathbf{0.35}$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0196$ 0.03 s.
ROC7	$K = 10^{-5}\begin{bmatrix} 0.0148 & -0.0016 & -0.0123 \\ 0.6947 & 0.0113 & 0.2792 \end{bmatrix}$ $\|K\|_2 = \mathbf{7.5\,10^{-6}}$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = 1.5\,10^{-7}$ 13 s.	$K = \begin{bmatrix} 0.5659 & 0.0379 & -0.1363 \\ -1.3274 & -0.9144 & 0.4762 \end{bmatrix}$ $\|K\|_2 = 1.76$ $\max_i \operatorname{Re}\lambda_i(A - BKC) = -0.0238$ 0.09 s.

obtained out of the four initializations (namely, identity for AC7 and AC8, random for HE1, and ABI for ROC7); see Table 6.2 for the results of the other initializations.

For these four numerical experiments, Algorithm 2 provides better solutions than RS-PHASE-II in 3 out of the 4 cases. In particular, for AC8 and ROC7, the improvement is significant; from $\|K\|_2 = 2.43$ to $\|K\|_2 = 0.02$ for AC8, and from $\|K\|_2 = 1.76$ to $\|K\|_2 = 7.5\,10^{-6}$ for ROC7.

Remark 3. For the experiment on ROC7 with Algorithm 2, the real part of one eigenvalue of $A - BKC$ is positive, namely $1.5\,10^{-7}$. The reason is that there is a numerical instability and limited precision when computing the eigenvalues of $A - BKC = (J - R)Q$. There are two ways to ensure that the eigenvalues of $A - BKC$ are within the left half of the complex plane.

A first possibility is to add ρI to the input matrix where ρ is a parameter, that is, apply Algorithm 2 on the matrix $A + \rho I$ (see Remark 2). Doing so with $\rho = 10^{-3}$ and using a random initialization provides

$$K = \begin{bmatrix} 0.2126 & -0.4960 & 0.2517 \\ 0.0929 & 0.1227 & 0.6055 \end{bmatrix}$$

with $\|K\|_2 = 0.6966$ where $\max_i \operatorname{Re}\lambda_i(A - BKC) = -5.6\,10^{-3}$.

A second possibility is to use a lower bound δ on the eigenvalues of R and Q which can be easily done in our code. For example, using $\delta = 10^{-6}$, we obtain $K = 10^{-3}\begin{bmatrix} 1.4 & 0 & 0 \\ 0 & 0 & 0.11 \end{bmatrix}$ with $\|K\|_2 = 1.4\,10^{-3}$ where $\max_i \operatorname{Re}\lambda_i(A - BKC) = -1.3\,10^{-17}$.

Table 6.2
Norms $\|K\|_2$ of the feedback matrices K (and, in brackets, the computational time in seconds) generated by Algorithm 2 using different initializations (second to fifth column) on problems from the COMPLeIB library. The last two columns report the results of the algorithm of Peretz (RS-PHASE-I=RS-P-I) [25, Table 8] and HIFOO [3, Table 1]. An empty box means that the result is not available, ∞ means that the algorithm did not find a feasible feedback, e-x means 10^{-x}. The solutions with error at most 0.01 away from the best solution found are highlighted in bold.

	Algorithm 2				RS-P.-I	HIFOO
	Identity	Random	ABI	AIC	[25]	[3]
AC1	**1.63e-9** (1.5)	**3.65e-9** (2.9)	**1.63e-9** (1.1)	**1.63e-9** (1.1)	1.62	**1.81e-15**
AC2	**1.63e-9** (1.3)	**3.65e-9** (3.3)	**1.63e-9** (1.1)	**1.63e-9** (1.1)	1.62	4.91e-2
AC4	7.48e-2 (2.1)	6.69e-2 (34)	7.27e-2 (8.0)	**5.21e-2** (15)	0.77	
AC5	1358.60 (16)	1344.56 (718)	2282.84 (28)	∞ (21)	**1.41**	1340.00
AC6	**4.05e-12** (1.1)	**4.05e-12** (3.3)	**4.05e-12** (1.1)	**4.05e-12** (1.1)	1.20	
AC7	**0.36** (6.5)	0.42 (68)	0.76 (21)	0.41 (59)	1.02	
AC8	**2.19e-2** (6.6)	0.12 (142)	1.20 (19)	0.50 (34)	2.43	
AC9	0.66 (7.9)	2.34 (137)	**1.31e-3** (36)	0.26 (44)	1.92	1.41
AC11	5.98 (11)	**0.17** (510)	1.30 (20)	0.92 (56)	2.41	3.64
AC12	1.79 (71)	1.19 (540)	1.00 (28)	0.90 (35)	3398.70	**5.00e-5**
AC15	**6.34e-10** (0.9)	**6.34e-10** (2.6)	**6.34e-10** (0.9)	**6.34e-10** (0.9)	8.71	
AC18	1.60 (223)	**0.37** (1688)	0.55 (46)	1.26 (173)	∞	18.60
HE1	2.26 (1.8)	0.42 (61)	0.68 (51)	0.46 (71)	1.67	**8.57e-2**
HE3	49.77 (26)	29.12 (589)	9.50 (136)	10.75 (140)	41.91	**0.81**
HE4	45.65 (22)	13.01 (569)	**5.68** (155)	9.42 (127)	145.17	18.60
HE5	∞ (38)	33.39 (1139)	∞ (180)	16.60 (38)	144.95	**1.59**
HE6	4.41 (15)	**2.30** (660)	2.47 (54)	9.83 (96)	3.63	∞
HE7	4.41 (15)	3.36 (439)	**2.47** (55)	9.83 (95)	3.63	∞
REA1	1.06 (2.6)	**0.85** (69)	0.94 (18)	1.79 (18)	1.55	1.50
REA2	1.73 (3.3)	**0.92** (45)	1.09 (15)	2.62 (16)	1.12	1.65
REA3	**2.40e-3** (0.0)	**2.40e-3** (30)	**2.40e-3** (0.0)	**2.40e-3** (0.0)	14.84	9.91
DIS2	6.18 (3.7)	4.94 (47)	8.59 (9.6)	3.24 (9.2)		**1.40**
DIS4	0.44 (70)	0.36 (227)	**0.32** (24)	0.40 (24)	2.82	1.69
DIS5	∞ (32)	**416.81** (503)	∞ (63)	∞ (30)	461.97	1280.00
WEC1	**4.30** (6.2)	25.98 (1365)	∞ (233)	9.05 (37)	∞	5.69
PAS	4.72e-3 (0.0)	4.72e-3 (13)	4.72e-3 (0.0)	4.72e-3 (0.0)	780.00	**1.97e-3**
TF1	1.16 (8.8)	4.00 (100)	0.21 (19)	∞ (126)	65.04	**0.14**
TF2	5.40 (3.2)	6.31 (154)	**0.42** (19)	1.69 (16)	7.95	10.90
TF3	∞ (97)	47.13 (656)	0.28 (19)	1.64 (31)	151.57	**0.14**
NN1	∞ (16)	48.43 (177)	∞ (43)	∞ (39)	133.69	**35.00**
NN2	**3.45e-9** (1.2)	**3.45e-9** (2.8)	**3.45e-9** (1.1)	**3.45e-9** (1.1)	1.35	1.54
NN3	∞ (31)	∞ (264)	∞ (37)	∞ (12)	∞	
NN5	∞ (30)	**17.14** (556)	∞ (73)	∞ (114)	39.03	82.40
NN6	∞ (115)	∞ (1446)	∞ (188)	∞ (264)	**110.73**	314.00
NN7	∞ (118)	∞ (1243)	∞ (187)	∞ (264)	**71.53**	84.20
NN9	∞ (39)	**10.66** (250)	∞ (41)	∞ (65)	504.50	20.90
NN12	32.67 (25)	18.23 (327)	∞ (88)	∞ (33)	27.05	**10.90**
NN13	0.61 (2.7)	0.40 (47)	**7.05e-2** (29)	9.98e-2 (17)	1.94	∞
NN14	0.61 (2.6)	0.38 (51)	**7.05e-2** (29)	9.98e-2 (17)	1.47	∞
NN15	**4.71e-11** (1.0)	**4.70e-11** (2.7)	**4.71e-11** (1.0)	**4.71e-11** (1.0)	2.00	4.80e-2
NN16	**1.52e-10** (1.1)	**1.83e-10** (3.1)	**1.52e-10** (1.0)	**1.52e-10** (1.0)	0.46	0.34
NN17	∞ (7.6)	53.23 (323)	1.86 (14)	**1.41** (11)	6.77	3.87
HF2D10	1.23 (2.1)	0.30 (31)	0.31 (3.7)	**0.29** (13)	15.41	70600
HF2D11	8.36 (2.1)	1.64 (32)	0.63 (3.6)	**0.58** (7.1)	44.02	85100
HF2D14	7.48e-2 (18)	2.37e-2 (97)	**2.06e-2** (4.8)	**2.04e-2** (9.8)		373000
HF2D15	0.91 (14)	0.26 (258)	**0.26** (14)	1.34 (103)		284000
HF2D16	2.98e-2 (3.0)	1.49e-2 (93)	1.57e-2 (4.4)	**1.39e-2** (4.2)		284000
HF2D17	0.10 (2.3)	**6.63e-2** (125)	6.81e-2 (12)	**6.59e-2** (11)		375000
HF2D18	2.45e-2 (2.1)	1.49e-2 (25)	1.84e-2 (7.1)	**7.00e-3** (15)		24.30
TMD	0.28 (2.8)	0.30 (88)	**2.15e-3** (7.7)	119.22 (37)	1.07	1.32
FS	∞ (35)	793.18 (382)	**145.34** (20)	1.26e+04 (38)		18300
ROC1	**5.63e-6** (0.0)	**5.63e-6** (11)	**5.63e-6** (0.0)	**5.63e-6** (0.0)	180.14	

Table 6.2 (*continued*)

	Algorithm 2				RS-P.-I	HIFOO
	Identity	Random	ABI	AIC	[25]	[3]
ROC2	∞ (54)	∞ (1471)	∞ (105)	∞ (185)	**152.94**	
ROC3	∞ (41)	∞ (642)	∞ (72)	∞ (107)	∞	
ROC4	**2.58e-6** (1.2)	**2.58e-6** (22)	**2.58e-6** (1.1)	**2.58e-6** (1.1)	241.57	
ROC5	**2.39e-9** (1.1)	**1.33e-9** (3.1)	**2.39e-9** (1.1)	**2.39e-9** (1.1)	232.22	
ROC7	0.18 (3.3)	0.37 (98)	**7.49e-6** (9.2)	3.01e-3 (15)	2.32	
# best	14 / 57	20 / 57	23 / 57	19 / 57	4 / 50	11 / 45
	Globally: 40 / 57					
# ∞	13 / 57	5 / 57	12 / 57	12 / 57	4 / 50	4 / 45
	Globally: 5 / 57					

When RS-PHASE-II obtains the best solution (HE1), the difference in norm is not significant: 0.42 for vs. 0.35. In summary, RE-PHASE-II is significantly faster than Algorithm 2, but generates in general solutions with larger norm. These observations will be confirmed in the next section.

6.2. Extensive numerical results

In this section, we report numerical results for all the systems tested in [25, Table 8] and [3, Table 1] of size $n \leq 20$; see Table 6.2. Note that some systems in [25, Table 8] were not tested in [3, Table 1], and vice versa, in which case we do not report any solution. Moreover, [25, Table 8] only reports the error of the initialization phase (RS-PHASE-I) hence we only report this result. Since [25, Table 8] minimize the ℓ_2 norm of K, we run Algorithm 2 using this norm.

In terms of solution quality, Algorithm 2 outperforms the two other approaches, providing the best solution in 40 out of the 57 cases (RS-PHASE-I does for 4 out of 50, HIFOO does for 11 out of 45). Moreover, in many cases, the ℓ_2 norm of the stabilizing feedback matrix is much smaller. Note however that HIFOO is not designed to minimize the ℓ_2 norm of K but rather to find a SOF that minimizes the H_∞/H_2-norm of the transfer function.

In terms of finding feasible solutions, the three algorithms are comparable: Algorithm 2 only fails 5 times out of 57, RS-PHASE-I 4 times out of 50, and HIFOO 4 times out of 45. All algorithms are sometimes able to find a feasible solution while the other fail. For example, RS-PHASE-I fails on AC18 and WEC1 while the two other algorithms succeed; HIFOO fails on HE6, HE7, NN13 and NN14; and Algorithm 2 fails on NN3-6-7-9 and ROC2-3.

Note that no algorithm is able to return a solution for NN3 and ROC3 hence these systems might not be stabilizable.

Sensitivity to initialization As expected, Algorithm 2 is rather sensitive to initialization. In terms of solution quality, ABI performs best (23 out of 57 best solution found). In terms of finding feasible solutions, Random performs best (only 5 failure out of 57) but used 10 initializations. In fact, we were curious to see whether the random initialization would be able to find feasible solutions if allowed more trials. We have rerun the same experiment with 100 initializations, and it was able to find a feasible solution for NN7 with $\|K\|_2 = 98.45$, and for ROC2 with $\|K\|_2 = 3.31$; both being better solutions than the ones found by RS-PHASE-I and HIFOO. Hence, except for NN6, Algorithm 2 with random initialization was able to find feasible solutions whenever RS-PHASE-I and HIFOO did.

Remark 4. In our experience, when Algorithm 2 fails, it means that the initial point does not allow to obtain a feasible point. In fact, increasing the number of iterations of the initialization phase (from 100 to 200) did not allow to obtain more feasible solutions with Algorithm 2.

Computational time RS-PHASE-I and -II require less than 1 seconds in all the examples shown on Table 6.2. Although it is not reported here, in terms of computational time, HIFOO is in general faster than Algorithm 2 but significantly slower than RS-PHASE-I and -II. For example, on NN9 ($n = 5$, $m = 3$ and $p = 2$), HIFOO requires about 6 seconds, while Algorithm 2 requires about 40 seconds on average. Note that the random initialization is slower than the other ones as it runs 10 times Algorithm 2.

7. Conclusion

In this paper, we have proposed a new characterization of all the SSFs and SOFs of a given LTI system pair (A, B) and system triplet (A, B, C), respectively, in terms of DH matrices. This allowed us to develop algorithms to compute minimal-norm SSFs for a system pair (A, B) (Algorithm 1) and minimal-norm SOFs for a system triplet (A, B, C) (Algorithm 2). Comparing Algorithm 2 with the methods HIFOO [3] and RS [25] on SOF problems, we found that RS and HIFOO performs better than Algorithm 2 in terms of computational time. In terms of solution quality, Algorithm 2 compares favorably with the two methods, being able to obtain better solution in many cases. In terms of finding feasible solutions, the three methods perform similarly.

Our characterization is very flexible as it can be directly extended to finding stabilizing feedbacks that achieve Ω-stability, where Ω is some specific LMI region; see Section 5.4. However, it currently does not scale well and can only be applied to medium-scale problems; see Section 5.3. An important direction of further research would therefore be to provide faster algorithms to tackle our optimization problems, for example based on first-order methods.

Declaration of competing interest

There is no competing interest.

Acknowledgements

The authors thank the anonymous referee for his/her insightful comments which helped improve the paper. The authors are also grateful to Yossi Peretz for sharing his code, and adressing our questions regarding his paper [25].

N. Gillis would like to thank Paul Van Dooren for his continuous support during his career. As a master student, Paul passed on to him the passion for matrix theory. Paul was then part of his Ph.D. committee during which Paul provided invaluable guidance. Paul also introduced him to the numerical linear algebra community (via meetings such as the first Gene Golub summer school and the Householder symposium); in particular Paul introduced him to Bob Plemmons with whom Nicolas had a fruitful collaboration.

Appendix A. Stabilizing matrix pairs (A, B)

In this section, we report the ℓ_2 norm of the solutions obtained by Algorithm 1 for the SSF problems corresponding to the same instance as in Table 6.2. We minimize the ℓ_2 norm of the feedback matrices, use $\underline{\epsilon} = 10^{-9}$, and update ϵ in the same way as in Section 6. We also report the error of the ℓ_2 norm of the solution obtained by solving (5.3) (initialization phase of Algorithm 1), and the error of the BCD algorithm that alternatively optimized (J, R) for Q fixed, and vice versa.

As mentioned in Section 5, SSDP performs significantly better than BCD in terms of solution quality. BCD provides a slightly better solution only in a few cases. In terms of computational time, BCD is faster as it solves subproblems with fewer variables; see Table A.1.

Table A.1
Comparison of the ℓ_2 norm of the stabilizing feedback matrices (and, in brackets, the computational time in seconds and the number of iterations) of BCD and SSDP for the SSF problem. The solutions with error at most 0.01% away from the best solution found are highlighted in bold, e-x means 10^{-x} and e+x means 10^{x}.

Data set (n, m)	Init.	BCD	SSDP
AC1 (5,3)	**2.82e-14** (1.0)	**2.82e-14** (0.0, 0)	**2.82e-14** (0.0, 0)
AC2 (5,3)	**2.82e-14** (0.3)	**2.82e-14** (0.0, 0)	**2.82e-14** (0.0, 0)
AC4 (4,1)	1.29 (1.2)	3.86e-1 (2.4, 4)	**7.91e-2** (10.4, 16)
AC5 (4,2)	1.43e+3 (0.8)	3.83e+2 (15.1, 23)	**2.85e+2** (280.4, 200)
AC6 (7,2)	**1.82e-15** (0.3)	**1.82e-15** (0.0, 0)	**1.82e-15** (0.0, 0)
AC7 (9,1)	2.33 (1.4)	2.09e-1 (2.8, 4)	**7.64e-2** (43.4, 25)
AC8 (9,1)	1.62 (1.5)	3.30e-1 (2.5, 4)	**3.79e-3** (37.5, 29)
AC9 (10,4)	1.13e-2 (1.7)	**3.06e-4** (7.0, 4)	4.02e-4 (53.9, 11)
AC11 (5,2)	2.44 (0.7)	1.49 (21.2, 23)	**7.44e-1** (19.2, 13)
AC12 (4,3)	4.73 (0.7)	**2.89** (6.3, 8)	4.00 (233.0, 200)
AC15 (4,2)	**2.33e-13** (0.3)	**2.33e-13** (0.0, 0)	**2.33e-13** (0.0, 0)
AC18 (10,2)	2.28e-1 (9.0)	3.21e-2 (23.4, 13)	**1.15e-2** (85.1, 21)
HE1 (4,2)	1.62e-1 (0.4)	1.26e-1 (3.3, 5)	**1.18e-1** (12.8, 12)
HE3 (8,4)	3.05 (1.9)	8.62e-1 (45.7, 35)	**7.06e-1** (205.5, 88)
HE4 (8,4)	1.45 (1.6)	3.28e-1 (65.3, 61)	**3.87e-2** (39.2, 16)
HE5 (8,4)	1.45 (1.3)	3.28e-1 (65.9, 61)	**3.87e-2** (38.2, 16)
HE6 (20,4)	1.45 (6.8)	7.83e-1 (553.4, 70)	**3.57e-2** (316.4, 19)
HE7 (20,4)	1.45 (6.8)	7.83e-1 (555.7, 70)	**3.57e-2** (317.1, 19)

(*continued on next page*)

Table A.1 (*continued*)

Data set (n, m)	Init.	BCD	SSDP
REA1 (4,2)	9.01e-1 (0.5)	3.70e-1 (4.4, 7)	**3.17e-1** (21.7, 20)
REA2 (4,2)	9.14e-1 (0.5)	3.75e-1 (5.8, 7)	**3.21e-1** (26.8, 25)
REA3 (12,1)	6.57e+2 (2.3)	6.12e+2 (3.7, 4)	**1.05e-2** (159.7, 96)
DIS2 (3,2)	2.60 (0.4)	1.41 (7.6, 12)	**1.22** (21.4, 23)
DIS4 (6,4)	7.19e-1 (0.5)	4.07e-1 (16.3, 21)	**3.20e-1** (21.7, 16)
DIS5 (4,2)	7.83e+2 (0.5)	1.58e+2 (1.3, 2)	**1.03e+2** (63.8, 39)
WEC1 (10,3)	3.27e+1 (7.3)	4.59 (261.5, 106)	**3.97** (72.3, 33)
PAS (5,1)	**6.82e-3** (0.9)	**6.82e-3** (0.0, 0)	**6.82e-3** (2.9, 1)
TF1 (7,2)	2.92e+1 (1.5)	3.97 (2.8, 4)	**7.77e-3** (30.2, 20)
TF2 (7,2)	2.92e+1 (1.6)	3.97 (2.8, 4)	**7.77e-3** (31.9, 20)
TF3 (7,2)	2.92e+1 (1.6)	3.97 (2.8, 4)	**7.77e-3** (30.4, 20)
NN1 (3,1)	1.65e+1 (0.8)	1.34e+1 (2.4, 4)	**1.30e+1** (22.9, 27)
NN2 (2,1)	**0** (0.1)	**0** (0.0, 0)	**0** (0.0, 0)
NN3 (4,1)	6.33e+1 (0.8)	4.31e+1 (0.6, 1)	**1.81e+1** (7.0, 6)
NN5 (7,1)	4.19e+2 (1.3)	1.69e+2 (2.9, 4)	**3.01e+1** (179.3, 200)
NN6 (9,1)	3.48e+2 (1.8)	3.12e+2 (0.7, 1)	**6.04e+1** (98.3, 60)
NN7 (9,1)	3.48e+2 (1.6)	3.12e+2 (0.7, 1)	**6.04e+1** (98.4, 60)
NN9 (5,3)	5.19 (0.6)	4.50 (2.2, 3)	**3.32** (66.4, 64)
NN12 (6,2)	2.45 (0.8)	2.45 (3.3, 4)	**1.41** (118.7, 111)
NN13 (6,2)	8.64e-1 (0.5)	8.34e-1 (2.6, 4)	**3.19e-1** (37.9, 35)
NN14 (6,2)	8.64e-1 (0.5)	8.34e-1 (2.6, 4)	**3.19e-1** (38.5, 35)
NN15 (3,2)	**2.22e-12** (0.3)	**2.22e-12** (1.3, 2)	**2.22e-12** (8.2, 1)
NN16 (8,4)	**5.40e-15** (0.3)	**5.40e-15** (0.0, 0)	**5.40e-15** (0.0, 0)
NN17 (3,2)	4.38 (0.7)	8.30e-1 (5.1, 8)	**7.26e-1** (10.2, 14)
HF2D10 (5,2)	9.37e-3 (0.5)	**9.16e-3** (2.1, 3)	9.19e-3 (1.5, 3)
HF2D11 (5,2)	1.55e-2 (0.5)	**1.50e-2** (2.9, 4)	**1.50e-2** (1.3, 3)
HF2D14 (5,2)	2.97e-2 (0.5)	**2.35e-2** (2.8, 4)	2.36e-2 (2.6, 5)
HF2D15 (5,2)	1.98e-1 (0.6)	**1.22e-1** (8.1, 12)	**1.22e-1** (6.0, 5)
HF2D16 (5,2)	1.46e-2 (0.5)	**1.40e-2** (2.7, 4)	**1.40e-2** (1.3, 3)
HF2D17 (5,2)	6.83e-2 (0.5)	**6.47e-2** (2.9, 4)	**6.47e-2** (2.4, 5)
HF2D18 (5,2)	3.79e-2 (0.5)	**3.59e-2** (1.3, 2)	3.79e-2 (5.0, 2)
TMD (6,2)	4.73e-3 (1.2)	**1.81e-4** (2.8, 4)	2.22e-4 (4.9, 8)
FS (5,1)	5.28e+2 (2.2)	5.28e+2 (0.0, 0)	**3.04e+1** (6.1, 8)
ROC1 (9,2)	**8.82e-5** (0.6)	**8.82e-5** (1.3, 1)	**8.82e-5** (35.6, 1)
ROC2 (10,2)	2.36 (2.9)	2.93e-1 (5.1, 4)	**7.78e-2** (121.4, 35)
ROC3 (11,4)	7.24 (1.7)	6.68 (12.8, 7)	**3.63** (92.2, 19)
ROC4 (9,2)	**1.79e-4** (0.6)	**1.79e-4** (2.2, 2)	**1.79e-4** (26.8, 1)
ROC5 (7,3)	**4.57e-14** (0.3)	**4.57e-14** (0.0, 0)	**4.57e-14** (0.0, 0)
ROC7 (5,2)	2.20 (1.8)	4.58e-1 (4.2, 7)	**1.33e-4** (8.6, 13)

References

[1] A. Albert, Conditions for positive and nonnegative definiteness in terms of pseudoinverses, SIAM J. Appl. Math. 17 (2) (1969) 434–440.

[2] B. Anderson, N. Bose, E. Jury, Output feedback stabilization and related problems – solution via decision methods, IEEE Trans. Autom. Control 20 (1) (1975) 53–66.

[3] D. Arzelier, G. Deaconu, S. Gumussoy, D. Henrion, H2 for HIFOO, in: International Conference on Control and Optimization with Industrial Applications, Bilkent University, Ankara, Turkey, 2011.

[4] K.J. Åström, R.M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, Princeton University Press, 2010.

[5] V. Blondel, J. Tsitsiklis, NP-hardness of some linear control design problems, SIAM J. Control Optim. 35 (6) (1997) 2118–2127.

[6] Y.Y. Cao, J. Lam, Y.X. Sun, Static output feedback stabilization: an ILMI approach, Automatica 34 (12) (1998) 1641–1645.

[7] N. Choudhary, N. Gillis, P. Sharma, On approximating the nearest Ω-stable matrix, in: Numerical Linear Algebra with Applications, 2019.

[8] A.R. Conn, N.I. Gould, P.L. Toint, Trust Region Methods, vol. 1, Siam, 2000.

[9] C.V.X. Research I.: CVX, Matlab software for disciplined convex programming, version 2.0, http://cvxr.com/cvx, 2012.
[10] N. Gillis, M. Karow, P. Sharma, Approximating the nearest stable discrete-time system, Linear Algebra Appl. 573 (2019) 37–53.
[11] N. Gillis, V. Mehrmann, P. Sharma, Computing nearest stable matrix pairs, Numer. Linear Algebra Appl. (2018) e2153, https://doi.org/10.1002/nla.2153.
[12] N. Gillis, P. Sharma, On computing the distance to stability for matrices using linear dissipative Hamiltonian systems, Automatica 85 (2017) 113–121.
[13] N. Gillis, P. Sharma, Finding the nearest positive-real system, SIAM J. Numer. Anal. 56 (2) (2018) 1022–1047.
[14] M. Grant, S. Boyd, Graph implementations for nonsmooth convex programs, in: Recent Advances in Learning and Control, 2008, pp. 95–110.
[15] L. Grippo, M. Sciandrone, On the convergence of the block nonlinear Gauss–Seidel method under convex constraints, Oper. Res. Lett. 26 (3) (2000) 127–136.
[16] S. Gumussoy, D. Henrion, M. Millstone, M.L. Overton, Multiobjective robust control with HIFOO 2.0, IFAC Proc. Vol. 42 (6) (2009) 144–149.
[17] D. Henrion, J. Lofberg, M. Kocvara, M. Stingl, Solving polynomial static output feedback problems with PENBMI, in: Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 7581–7586.
[18] T. Iwasaki, R. Skelton, All controllers for the general H_∞ control problem: LMI existence conditions and state space formulas, Automatica 30 (8) (1994) 1307–1317.
[19] T. Iwasaki, R. Skelton, J. Geromel, Linear quadratic suboptimal control with static output feedback, Syst. Control Lett. 23 (6) (1994) 421–430.
[20] F. Leibfritz, Compleib, constraint matrix-optimization problem library-a collection of test examples for nonlinear semidefinite programs, control system design and related problems, Dept. Math., Univ. Trier, Trier, Germany, 2004, Tech. Rep.
[21] C. Mehl, C. Mehrmann, P. Sharma, Stability radii for linear Hamiltonian systems with dissipation under structure-preserving perturbations, SIAM J. Matrix Anal. Appl. 37 (4) (2016) 1625–1654.
[22] M. Mesbahi, A semi-definite programming solution of the least order dynamic output feedback synthesis problem, in: Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304), vol. 2, 1999, pp. 1851–1856.
[23] A. Nemirovskii, Several NP-hard problems arising in robust stability analysis, Math. Control Signals Syst. 6 (2) (1993) 99–105.
[24] Y. Peretz, A characterization of all the static stabilizing controllers for LTI systems, Linear Algebra Appl. 437 (2) (2012) 525–548.
[25] Y. Peretz, A randomized approximation algorithm for the minimal-norm static-output-feedback problem, Automatica 63 (2016) 221–234.
[26] B. Polyak, M. Khlebnikov, P. Shcherbakov, An LMI approach to structured sparse feedback design in linear control systems, in: 2013 European Control Conference (ECC), 2013, pp. 833–838.
[27] B.F. Spencer, M.K. Sain, Controlling buildings: a new frontier in feedback, IEEE Control Syst. 17 (6) (1997) 19–35.
[28] J.G. Sun, Backward perturbation analysis of certain characteristic subspaces, Numer. Math. 65 (1993) 357–382.
[29] V.L. Syrmos, C.T. Abdallah, P. Dorato, K. Grigoriadis, Static output feedback: a survey, Automatica 33 (2) (1997) 125–137.
[30] K.C. Toh, M. Todd, R. Tütüncü, SDPT3–a MATLAB software package for semidefinite programming, version 1.3, Optim. Methods Softw. 11 (1–4) (1999) 545–581.
[31] H.L. Trentelman, A.A. Stoorvogel, M. Hautus, Control Theory for Linear Systems, Springer Science & Business Media, 2012.
[32] R. Tütüncü, K. Toh, M. Todd, Solving semidefinite-quadratic-linear programs using SDPT3, Math. Program. 95 (2) (2003) 189–217.
[33] Y. Xu, J. Teng, Optimum design of active/passive control devices for tall buildings under earthquake excitation, Struct. Des. Tall Spec. Build. 11 (2) (2002) 109–127.
[34] J.N. Yang, S. Lin, F. Jabbari, H2-based control strategies for civil engineering structures, J. Struct. Control. 10 (3–4) (2003) 205–230.