Feedback Control for Structured Descriptor Systems with Minimum Free-entry Pattern Gain Vectors ⋆

Subashish Datta [a], Harsha Mathur [b]

[a]*Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, India-110016.*

[b]*R&D Industry, Bengaluru, India.*

Abstract

In this article, a *structured descriptor system* (SDS) is considered, where the entries of the system matrices are either fixed zeros or free parameters. When the free-entries are assigned with some real numbers, the resulting system becomes a descriptor system in the usual sense. The objective of this work is to devise a methodology to verify whether there exists a static state feedback control for a SDS such that almost all the DSs (in a generic sense), characterized by the given SDS, can be made impulse free and place their finite poles at any desired locations in the complex plane. In addition, if such a feedback control exists, then compute a structured gain vector, which has minimum free-entries. To address the underlying problems, graph theoretic approach is used, where it is shown that the computation of spanning cycle families of a directed graph, corresponding to a suitably defined structured matrix, plays an important role. The efficacy of the developed results are demonstrated with a numerical example.

Key words: Descriptor systems, structured matrices, graph theory, state feedback control.

1 Introduction

The combination of differential and algebraic equations, to model various physical systems, such as power networks (Sauer & Pai 1998), biological systems (Liu et al. 2008) and cyber-physical systems (Pasqualetti et al. 2013), is common in practice. Such a mathematical model, when linearized, is represented in the descriptor form: $E\dot{x} = Ax + Bu$, where the matrix E is singular, and the associated system is referred to as *descriptor system* (DS). The response of a DS may contain impulses and its derivatives, which is possibly due to the presence of inconsistent initial conditions and/or non-smoothness of (control) input to the system (Verghese et al. 1981, Dai 1989, Duan 2010). Such impulsive response is undesirable in the practical applications, since its presence may saturate or destroy the physical components. Moreover, it is often required that the system response should satisfy transient performance specifications in order to ensure safe operation. These problems are well-addressed, when the exact numerical values of entries of the system matrices (E,A,B) are known. In particular, static state feedback control has been proposed in Dai (1989), Duan (2010), Datta (2017) to: i) make the *(nilpotent) index* of closed loop system one for impulse elimination and ii) place the finite poles at some fixed locations or within a region in the complex plane for achieving specified transient response.

In this work, we consider the impulse elimination and finite pole assignment problems for a SDS: $\overline{E}\dot{x} = \overline{A}x + \overline{B}u$, where the only known information about system matrices $(\overline{E},\overline{A},\overline{B})$ is their structure, that is, the entries are either fixed zeros or free parameters (can take any real numbers). When the free-entries of $\overline{E}$, $\overline{A}$ and $\overline{B}$ are assigned with some particular real numbers, the resulting system becomes a *numerical realization* of SDS or simply a DS. Hence, a SDS characterizes a class of DSs that are having same structure (zero/free-entries pattern) in the system matrices. In this work, we consider a single input SDS, and address the following problems: i) determine if there exist a static state feedback control for a given SDS such that almost all (in a generic sense (Dion et al. 2003)) the numerical realizations of closed-loop SDS can be made impulse-free and assign their finite poles at any desired locations in the complex plane, and ii) if such a feedback control exists, then determine the number of minimum free-entries required in the gain vector. The contributions to address these problems are as follows.

- For a given SDS with static state feedback control, we define two structured matrices: $\overline{H}_c$ and $\overline{H}_o$, and associate the

⋆ This work is partially supported by the Science and Engineering Research Board (SERB), Government of India, India, Grant Number: CRG/2021/003783.

Email addresses: subashish@ee.iitd.ac.in (Subashish Datta), harshamathur92@gmail.com (Harsha Mathur).

following *directed graphs* (digraphs): $\mathscr{G}(\overline{H}_c)$ and $\mathscr{G}(\overline{H}_o)$ with them. Then, we show that the coefficients of closed loop characteristic polynomial can be determined from the spanning cycle families (SCFs) of $\mathscr{G}(\overline{H}_c)$.
- Using the digraph $\mathscr{G}(\overline{H}_o)$, we provide a procedure to design a minimum free-entries gain vector such that the closed loop SDS is structurally impulse-free.
- By defining a structured matrix $\overline{Z}$, an algebraic relation between the coefficients of characteristic polynomials, feedback gain vector and $\overline{Z}$ is derived. We prove, using the concept of *1-connection* of a digraph (Brualdi & Cvetkovic 2009), that the elements of $\overline{Z}$ can be determined from the SCFs of $\mathscr{G}(\overline{H}_o)$.
- We show that the existence of a feedback gain vector, for achieving the objectives of impulse elimination and finite pole assignment (in a generic sense) in a SDS, can be determined from the structural rank of $\overline{Z}$. We associate a bipartite graph $\mathscr{G}_b(\overline{Z})$ with $\overline{Z}$, and then show that a minimum free-entries feedback gain vector can be computed by finding a *maximum matching* of $\mathscr{G}_b(\overline{Z})$.
- Since the computation of SCFs of a digraph plays an important role, we provide algorithms to obtain all possible SCFs of a digraph, and the associated vertex disjoint cycles, by exploiting the properties of incidence matrix.

The structural properties and solvability of control problems for structured systems, particularly for ordinary state space systems and the corresponding transfer function matrices, have been studied extensively in the literature (see Lin (1974), Reinschke (1988), Murota (1987*b*), Woude & Murota (1995), Dion et al. (2003), Moothedath et al. (2018), Blanchini & Giordano (2021), Ramos et al. (2022) and the references therein). Although, a most commonly adopted tool to address the underlying problems is graph theory, some of the available work use the concept of *matroid* (Murota 1987*b*). The digraph representation of a SDS and the structural properties, such as impulse controllability, $\mathscr{R}$-controllability, $\mathscr{C}$-controllability and $\mathscr{R}$-observability are studied in (Yamada & Luenberger 1985, Murota 1987*a*, Reinschke 1994, Reinschke & Wiedemann 1997, Boukhobza et al. 2006). The problem of causality/impulse freeness of a networked descriptor system is considered in Zhou (2022). Further, Clark et al. (2017) have developed algorithms, in the SDS framework, for controlling a network of systems. Despite these developments, the problem of impulse elimination and finite pole assignment for SDS seems to be unavailable in the literature. A preliminary work, related to finite pole assignment for small scale (two or three states) SDSs, is presented in Mathur & Datta (2018). These results are improved in this work significantly, and included the objectives of impulse elimination and computation of a minimum free-entry pattern gain vector.

The remaining part of this article is organized as follows. Preliminaries on SDSs and problem formulation are mentioned in Section 2. Main results and the relevant algorithms are presented in Section 3 and Section 4, respectively. The concluding remarks are mentioned in Section 6, following to a demonstrative example in Section 5.

2 Preliminaries and Problem Formulation

Consider a set of structured matrices (see Appendix-A for the definition): $\overline{E}$, $\overline{A}$ and $\bar{b}$ of sizes $n\times n$, $n\times n$ and $n\times 1$, respectively. Then, a single-input SDS is represented by:

$$\overline{E}\dot{x} = \overline{A}x + \bar{b}u, \tag{1}$$

where $\overline{E}$ is structurally singular, x is the state vector and u is the (control) input to the system. Let e_{ij}, a_{ij} and b_{i1}, for $i,j = 1,2,\cdots,n$, be the entries of $\overline{E}$, $\overline{A}$ and $\bar{b}$, respectively. Denote the matrices E, A and b as numerical realizations (see Appendix-A for the definition) of $\overline{E}$, $\overline{A}$ and $\bar{b}$, respectively. Then, a *numerical realization* of SDS (1) is represented as: $E\dot{x} = Ax + bu$. Throughout this article, it is assumed that the *structural rank* of $\overline{E}$ (s-rank$(\overline{E})$) is r. Then, by definition, the numerical rank of almost all E is r. Corresponding to the SDS (1), define a structured matrix: $\overline{G}(s) := \big(s\overline{E} - \overline{A}\big)$, where s can take any values from the set of complex numbers. Further, define a (multivariate) polynomial: $\overline{\alpha}(s) := \det\big(\overline{G}(s)\big)$, where det$(\bullet)$ stands for determinant of a matrix, and represent it as follows:

$$\overline{\alpha}(s) = \bar{\alpha}_l s^l + \bar{\alpha}_{l-1}s^{l-1} + \ldots + \bar{\alpha}_1 s + \bar{\alpha}_0, \tag{2}$$

with coefficients $\bar{\alpha}_k$, for $k = 0,1,\cdots,l$, being the functions of variables e_{ij} (free-entries of $\overline{E}$) and a_{ij} (free-entries of $\overline{A}$). We say that system (1) is *structurally regular* if $\overline{G}(s)$ is structurally non-singular, that is, $\overline{\alpha}(s)$ is not identically equal to 0. In the remaining parts of this article, we assume that SDS (1) is structurally regular. Then, for almost all E and A, the polynomial: $\alpha(s) := \det(sE - A) \not\equiv 0$, and hence, the corresponding numerical realizations of (1) are regular (Dai 1989, Duan 2010). We refer to the roots of $\alpha(s)$ as *finite poles* of a particular numerical realization ($E\dot{x} = Ax + bu$) of (1). In (2), if $\bar{\alpha}_l \neq 0$, then the degree of polynomial $\alpha(s)$, for almost all E and A, is equal to l, and hence, we refer to l as *structural degree* of $\overline{\alpha}(s)$ (also see Reinschke (1994) for the definition of *generic degree*).

It is well known (Dai 1989) that the matrix pair (E,A), associated with every (regular) numerical realizations of (1), can be transformed into *Weierstrass canonical form*, that is, there exist two nonsingular matrices $S \in \mathbb{R}^{n\times n}$ and $T \in \mathbb{R}^{n\times n}$ such that $SET = \left[\begin{smallmatrix} I & \mathbf{0} \\ \mathbf{0} & N \end{smallmatrix}\right]$ and $SAT = \left[\begin{smallmatrix} J & \mathbf{0} \\ \mathbf{0} & I \end{smallmatrix}\right]$, where $J \in \mathbb{R}^{r\times r}$ and $N \in \mathbb{R}^{(n-r)\times(n-r)}$ is a nilpotent matrix. The index ρ of N ($N^{\rho} = 0$ and $N^{\rho-1} \neq 0$) is referred to as *index* of $E\dot{x} = Ax + bu$. In the Weierstrass canonical form, if $N = \mathbf{0}$, then the (distributional) solution of $E\dot{x} = Ax + bu$ is free from impulses and its derivatives (if $\rho \geq 2$, then the solution is impulsive) (Dai 1989, Duan 2010). Another necessary and sufficient condition, which ensures impulse-free solution of $E\dot{x} = Ax + bu$, is that the degree of $\alpha(s)$ is equal to (numerical) rank$(E) = r$ (Duan 2010, Theorem 7.1). Now, if we assume that the structural degree of $\overline{\alpha}(s)$ is equal to s-rank$(\overline{E})$, that is, $l = r$, then for almost all numerical realizations of (1), the degree of $\alpha(s)$ is equal to r, and hence, their solutions are free from impulses and its derivatives. We use the following definition.

Definition 1 *A SDS* (1) *is said to be structurally impulse-free (SIF) if the structural degree of* $\overline{\alpha}(s)$ *is equal to s-rank*$(\overline{E})$*, that is,* $l = r$.

Consider a linear static state feedback control of the form:

$$u = \bar{f}^T x, \tag{3}$$

where $\bar{f} = [f_{11}\ f_{12}\ \cdots\ f_{1n}]^T$, and f_{1j} is a free-entry of $\bar{f}$. Then, the closed loop system, corresponding to SDS (1), is:

$$\overline{E}\dot{x} = \left(\overline{A} + \bar{b}\bar{f}^T\right)x. \tag{4}$$

A numerical realization of closed loop SDS (4) is represented as: $E\dot{x} = \left(A + bf^T\right)x$, where f is a numerical realization of $\bar{f}$. Corresponding to the SDS (4), define the following (multivariate) polynomial: $\overline{\sigma}(s) := \det(s\overline{E} - \overline{A} - \bar{b}\bar{f}^T)$, and represent it as follows:

$$\overline{\sigma}(s) = \bar{\sigma}_q s^q + \bar{\sigma}_{q-1}s^{q-1} + \cdots + \bar{\sigma}_1 s + \bar{\sigma}_0, \tag{5}$$

where $\bar{\sigma}_k$, for $k = 0, 1, \cdots, q$, are functions of the variables $e_{ij}, a_{ij}, b_{i1}, f_{1j}$, which are free-entries of $\overline{E}$, $\overline{A}$, $\bar{b}$ and $\bar{f}$, respectively. According to Definition 1, the closed loop SDS (4) is SIF, if $q = \text{s-rank}(\overline{E}) = r$, where q is the structural degree of $\overline{\sigma}(s)$. The roots of the polynomial: $\sigma(s) := \det(sE - A - bf^T)$ are referred to as finite poles of a particular numerical realization of (4). Then, the problem of interest is posed as follows, where the following control objective is considered.

Objective 1 *The closed loop SDS* (4) *is SIF and the finite poles of almost all of its numerical realizations can be assigned at any desired locations in the complex plane.*

Problem 2 *For a given SDS* (1)*, determine if there exists a feedback gain vector* $\bar{f}$ *such that Objective 1 is achieved. Further, if such a gain vector exists, then compute a structured gain vector* $\bar{f}^*$*, having minimum number of free-entries, such that Objective 1 is achieved.*

Since the entries of the system matrices: $\overline{E}$, $\overline{A}$ and $\bar{b}$ in (1) do not have particular numerical values, and the only known information is their zero/free-entry pattern, we use graph theoretic approach to address Problem 2. We first derive a condition for the existence of gain vector $\bar{f}$, and then, we give a procedure to obtain $\bar{f}^*$. More details on graph theoretic definitions and results are presented in Appendix - A.

3 Main Results

Corresponding to the SDS (1) with feedback control (3), define a structured matrix of size $(n+1) \times (n+1)$ as follows: $\overline{H}_c := \begin{bmatrix} s\overline{E}-\overline{A} & \bar{b} \\ \bar{f}^T & 1 \end{bmatrix}$. Then, similar to Reinschke (1994), we associate a digraph $\mathscr{G}(\overline{H}_c)$, with the structured matrix $\overline{H}_c$, which has $n+1$ vertices: $V := \{v_1, v_2, \cdots, v_{n+1}\}$ and following edges: i) from vertex v_j to v_i with weight $-a_{ij}$, if a_{ij} is a free-entry of $\overline{A}$ (referred to it as A-edge), ii) from vertex v_j to v_i with weight se_{ij}, if e_{ij} is a free-entry of $\overline{E}$ (referred to it as E-edge), iii) from vertex v_{n+1} to v_i with weight b_{i1}, if b_{i1} is a free-entry of $\bar{b}$ (referred to it as b-edge), iv) from vertex v_j to v_{n+1} with weight f_{1j}, (referred to it as f-edge), and v) a self-loop at vertex v_{n+1} with weight 1. Note that if both e_{ij} and a_{ij} are free-entries in $\overline{E}$ and $\overline{A}$, respectively, then $\mathscr{G}(\overline{H}_c)$ has two parallel edges from vertex v_j to v_i: i) one has edge weight se_{ij} and ii) another has edge weight $-a_{ij}$. To distinguish E-edges and A-edges of $\mathscr{G}(\overline{H}_c)$, we use dashed lines and solid lines for E-edges and A-edges, respectively. The other edges are represented by solid lines (refer to Example 1). The digraph $\mathscr{G}(\overline{G}(s))$, associated with the structured matrix $\overline{G}(s)$, is also constructed in the similar way, and hence, same notations are used, as it is used for $\mathscr{G}(\overline{H}_c)$. We now show that the coefficients $\bar{\sigma}_k$'s of $\overline{\sigma}(s)$, as in (5), can be obtained from the SCFs of $\mathscr{G}(\overline{H}_c)$.

Lemma 3 *Let* $\overline{\sigma}(s)$ *be as in* (5)*. For* $k = 0, 1, \cdots, q$*, let* $C_s(E_k)$ *be a SCF of* $\mathscr{G}(\overline{H}_c)$*, which has exactly* k *E-edges. Denote* $w(C_s(E_k))$ *as the product of weights of the edges in* $C_s(E_k)$*, excluding the term* s *that appears in E-edges. For a fixed* k*, let* $\mathscr{C}_k$ *be the set of all* $C_s(E_k)$*'s in* $\mathscr{G}(\overline{H}_c)$ *and* δ *be the number of vertex disjoint cycles in* $C_s(E_k)$*. Then,* $\sigma_k = \sum_{\mathscr{C}_k}(-1)^{n+1-\delta}w(C_s(E_k))$.

Proof 1 Note that $\overline{\sigma}(s) = \det(\overline{H}_c)$. It then follows from (Reinschke 1988, Theorem A2.1) that each summand of $\det(\overline{H}_c)$ corresponds to a SCF of digraph $\mathscr{G}(\overline{H}_c)$, and its value is equal to the product of weight of the edges involved in that SCF. Further, since the size of $\overline{H}_c$ is $(n+1) \times (n+1)$, the sign factor of each summand is determined by the relation: $(-1)^{n+1-\delta}$. Now, consider a SCF $C_s(E_k)$. Since the weight of an E-edge in $C_s(E_k)$ is se_{ij} and $C_s(E_k)$ has exactly k E-edges, the product of weights of the edges in $C_s(E_k)$ is $s^k w(C_s(E_k))$. Hence, $\sigma_k s^k$ can be computed by collecting all $C_s(E_k) \in \mathscr{C}_k$. In particular, for a fixed k, the addition of all $w(C_s(E_k))$ (with appropriate sign factor), corresponding to the SCFs in $\mathscr{C}_k$, results in σ_k.

3.1 Design of $\bar{f}$ for Impulse Elimination

We now give a procedure to design a structured gain vector $\bar{f}$ to make the closed loop SDS (4) SIF. For this, following square structured matrix of size $(n+1) \times (n+1)$ is defined:

$$\overline{H}_o := \begin{bmatrix} s\overline{E} - \overline{A} & \bar{b} \\ \bar{f}^T & 0 \end{bmatrix}. \tag{6}$$

Observe that $\overline{H}_o$ is same as $\overline{H}_c$, except the entry $\overline{H}_c(n+1, n+1)$, which is changed to 0. Let $\mathscr{G}(\overline{H}_o)$ be the digraph associated with $\overline{H}_o$. Then, all the edges and vertices of $\mathscr{G}(\overline{H}_c)$ will appear in $\mathscr{G}(\overline{H}_o)$, except the self-loop at vertex v_{n+1}. Hence, we use same edge names for $\mathscr{G}(\overline{H}_o)$, which are used for $\mathscr{G}(\overline{H}_c)$. Assume that the open loop SDS (1) is not SIF, that is, the structural degree of $\overline{\alpha}(s)$ is strictly less than s-rank($\overline{E}$) ($l < r$). It is shown in Reinschke & Wiedemann (1997) that SDS (1) is structurally impulse controllable, that is, almost all numerical realizations of (1) are impulse controllable if and only if the digraph $\mathscr{G}(\overline{H}_o)$ has a SCF involving r E-edges. Further, it follows from (Duan 2010, Theorem 7.6) that the impulse controllable numerical realizations of (1) can be made impulse-free with a feedback control of the form (3). We now give a procedure to design a structured gain vector $\bar{f}$, which has minimum number of free-entries, to make the closed loop (4) SIF. Before proceeding to the main result, we consider the following proposition.

Proposition 4 *No SCF of $\mathscr{G}(\overline{H}_c)$ can have more than r E-edges, where s-rank($\overline{E}$) $= r$.*

Proof 2 Consider a SCF of $\mathscr{G}(\overline{H}_c)$, which contains maximum number of E-edges, and denote it as $C_s(E_h)$. Assume that $C_s(E_h)$ has $h > r$, E-edges. Now, delete (if they exist) A-edges, b-edges, f-edges, the self-loop at vertex v_{n+1} and vertex v_{n+1} from $C_s(E_h)$. Then, the resulting digraph $\widetilde{C}_s$ contains only E-edges. Since these E-edges are part of vertex disjoint cycles in $C_s(E_h)$, at each vertex of $\widetilde{C}_s$, at most one edge either enters or leaves. Note that $\widetilde{C}_s$ is a spanning subdigraph of $\mathscr{G}(\overline{E})$, which contains $h > r$, E-edges. Hence, it follows from Proposition 10 (see Appendix-A) that s-rank($\overline{E}$) $> r$, which is a contradiction. Hence, the proposition holds.

Theorem 5 *Let $C_s(E_r)$ be a SCF of $\mathscr{G}(\overline{H}_o)$, which has r E-edges. Then, the following statements hold.*

(1) The SCF $C_s(E_r)$ contains only one b-edge and one f-edge, and it is also a SCF of $\mathscr{G}(\overline{H}_c)$.
(2) The gain vector, which makes the closed loop SDS (4) SIF and has minimum free-entries, is of the following structure: $\bar{f} = [0 \ \cdots \ 0 \ f_{1k} \ 0 \ \cdots \ 0]$, where f_{1k} is the weight of f-edge that appears in $C_s(E_r)$.

Proof 3 Define a set $\mathscr{N} := \{1, 2, \cdots, n\}$. In $\mathscr{G}(\overline{H}_o)$, every b-edge leaves from v_{n+1} and enters at v_i, for $i \in \mathscr{N}$, and every f-edge leaves from v_k and enters at v_{n+1}, for $k \in \mathscr{N}$. Since $\mathscr{G}(\overline{H}_o)$ does not have self-loop at vertex v_{n+1}, every SCF of $\mathscr{G}(\overline{H}_o)$ must contains both b-edge and f-edge. Further, according to the definition of SCF, only one edge can enter and one edge can leave at each vertex of a SCF. Hence, $C_s(E_r)$ contains only one b-edge and one f-edge. According to the definitions of $\overline{H}_c$ and $\overline{H}_o$, it is easy to notice that a SCF of $\mathscr{G}(\overline{H}_o)$ is also a SCF of $\mathscr{G}(\overline{H}_c)$. Hence, the first statement holds.

According to Definition 1, the closed loop SDS (4) is SIF if the structural degree of $\overline{\sigma}(s)$ is r, that is, $q = r$. Since $C_s(E_r)$ is also a SCF of $\mathscr{G}(\overline{H}_c)$, it follows from Lemma 3 that $\sigma_r \neq 0$. It is shown in Proposition 4 that none of the SCFs of $\mathscr{G}(\overline{H}_c)$ can have more than r E-edges, since s-rank($\overline{E}$) $= r$. Hence, the structural degree of $\overline{\sigma}(s)$ is r, and the closed loop SDS (4) is SIF. It then follows from the first statement that the minimum number of free-entries required in the gain vector $\bar{f}$, to make (4) SIF, is one. Assume that $C_s(E_r)$ contains the f-edge that leaves from vertex v_k and enters at vertex v_{n+1} with edge weight f_{1k}, for $k \in \mathscr{N}$. Then, the feedback gain vector $\bar{f}$ that makes the closed loop SDS (4) SIF is: $\bar{f} = [0 \cdots 0 \ f_{1k} \ 0 \cdots 0]$. Note that $\mathscr{G}(\overline{H}_o)$ may contain more than one SCFs having r E-edges, however, to make (4) SIF, we need to consider only those SCFs where f_{1k} appears. This completes the proof.

It is shown in Pasqualetti et al. (2013) and Chowdhury et al. (2020) that a cyber-physical system (CPS) can be modeled as a DS. Hence, if an impulsive structured CPS is made impulse-free using a feedback control of the form (3), then it is required to identify the *critical feedback path* f_{1k}, whose absence can make the system impulsive. Theorem 5 (statement 2) is useful to determine such critical feedback paths. It is important to provide special security to secure these critical feedback paths from the cyber-attacks.

3.2 Condition for the existence of $\bar{f}$ and design of $\bar{f}^*$ to achieve Objective 1

In this section, we derive a condition for the existence of $\bar{f}$ and propose a procedure to obtain a structured $\bar{f}^*$ for achieving Objective 1. According to Definition 1, the closed loop SDS (4) is SIF if $q =$ s-rank($\overline{E}$) $= r$, where q is the structural degree of $\overline{\sigma}(s)$. Assume that: i) the structural degree of $\overline{\sigma}(s)$ is r, that is, $q = r$ and open loop SDS (1) is not SIF, that is, $l < r$ (when (1) is SIF, $l = r$). Then, corresponding to $\overline{\alpha}(s)$ and $\overline{\sigma}(s)$, as represented in (2) and (5), respectively, define the following two coefficient vectors:

$$\overline{\alpha} := \begin{bmatrix} \bar{\alpha}_0 & \cdots & \bar{\alpha}_l & \mathbf{0}_{r-l} \end{bmatrix}^T, \ \overline{\sigma} := \begin{bmatrix} \bar{\sigma}_0 & \cdots & \bar{\sigma}_{r-1} & \bar{\sigma}_r \end{bmatrix}^T, \quad (7)$$

where $\mathbf{0}_{r-l}$ is a vector of size $r-l$ with entries 0. Define:

$$\overline{D}(s) := \text{adj}\left(\overline{G}(s)\right) = \overline{D}_r s^r + \cdots + \overline{D}_1 s + \overline{D}_0, \quad (8)$$

where adj(.) denotes the adjoint of a matrix, and $\overline{D}_k$'s, for $k = 0, 1, \cdots, r$, are the coefficient matrices of polynomial matrix $\overline{D}(s)$. In the following result we propose how to compute the elements of $\overline{D}_k$ using 1-*connections* of the digraph $\mathscr{G}(\overline{G}(s))$ (see Appendix-A for the definition). For this, following notations are used: i) L_{ij}: 1-connection of $\mathscr{G}(\overline{G}(s))$ from vertex v_j to vertex v_i, ii) L^k_{ij}: 1-connection of $\mathscr{G}(\overline{G}(s))$ from v_j to v_i, which contains exactly k E-edges, iii) $w(L_{ij})$: the product of weight of the edges in L_{ij}, and iv) $w'(L^k_{ij})$: the product of weight of the edges in L^k_{ij} by dropping the term s that appears in E-edges.

Lemma 6 *For $i, j = 1, 2, \cdots, n$ and $k = 0, 1, \cdots, r$, denote the $(i, j)^{th}$ element of coefficient matrix $\overline{D}_k$ as d^k_{ij}. Then, $d^k_{ij} = \sum_{\mathscr{L}^k_{ij}} (-1)^{n+\bar{\delta}+1} w'(L^k_{ij})$, where $\bar{\delta}$ is the number of vertex disjoint cycles in L^k_{ij}, and $\mathscr{L}^k_{ij}$ is the set of all L^k_{ij}'s.*

Proof 4 Let $c_{ji}(s)$ be the cofactor of an element of $\overline{G}(s)$. Then, $c_{ji}(s) = (-1)^{i+j}\det((\overline{G}(s))_{ji})$, where $(\overline{G}(s))_{ji}$ is a matrix obtained from $\overline{G}(s)$ by deleting the j^{th} row and i^{th} column. By denoting $d_{ij}(s)$ as $(i,j)^{th}$ element of $\overline{D}(s)$, we have $d_{ij}(s) = c_{ji}(s)$. It then follows from (Brualdi & Cvetkovic 2009, Chapter 5) that: $d_{ij}(s) = c_{ji}(s) = \sum_{\mathscr{L}_{ij}}(-1)^{n+\bar{\delta}+1}w(L_{ij})$, where $\bar{\delta}$ is the number of vertex disjoint cycles in L_{ij}, and $\mathscr{L}_{ij}$ is the set of 1-connections L_{ij}. The sign factor of $w(L_{ij})$ is determined by the relation $(-1)^{n+\bar{\delta}+1}$. It can be observed that $w(L_{ij}) = s^k w'(L^k_{ij})$, if L_{ij} contains exactly k E-edges. According to the definition, L_{ij} (for $i \neq j$) is a spanning subdigraph of $\mathscr{G}(\overline{G}(s))$, and it contains: i) vertex disjoint cycles and a path or ii) only a path from vertex v_j to vertex v_i. Similarly, L_{jj} contains vertex disjoint cycles and an isolated vertex v_j. Using these facts, it can be shown (similar to the proof of Proposition 4) that the maximum number of E-edges that can appear in L_{ij} is r. Hence, it follows that $d_{ij}(s)$ can be represented as: $d_{ij}(s) = \sum_{k=0}^{r} d^k_{ij}s^k$, where the coefficient d^k_{ij} can be obtained using 1-connections L^k_{ij}. For some fixed i, j and k, there might be multiple L^k_{ij}'s, which form the set $\mathscr{L}^k_{ij}$. Hence, d^k_{ij} is the addition of all $w'(L^k_{ij})$'s with sign factor $(-1)^{n+\bar{\delta}+1}$ for individual summand.

Theorem 7 *Let $\overline{\alpha}$ and $\overline{\sigma}$ be as in* (7)*, and $\overline{D}(s)$ be as in* (8)*. Then, by defining a matrix $\overline{Z} := \left[\overline{D}_0\bar{b}\ \overline{D}_1\bar{b}\ \cdots\ \overline{D}_r\bar{b}\right]$ of size $n\times(r+1)$, following relation holds:*

$$\overline{Z}^T\bar{f} = \overline{\alpha} - \overline{\sigma}. \tag{9}$$

Proof 5 Recall that SDS (1) is structurally regular, and hence, $\left(\overline{G}(s)\right)^{-1}$ exists. Further, noticing the fact that the polynomial: $\overline{\sigma}(s) = \det\left(\overline{G}(s) - \bar{b}\bar{f}^T\right)$, we have:

$$\begin{aligned}\overline{\sigma}(s) &= \det\left[\left(\overline{G}(s)\right)\left\{I - \left(\overline{G}(s)\right)^{-1}\bar{b}\bar{f}^T\right\}\right]\\ &= \overline{\alpha}(s)\left(1 - \bar{f}^T\left(\overline{G}(s)\right)^{-1}\bar{b}\right)\\ &= \overline{\alpha}(s) - \bar{f}^T\operatorname{adj}\left(\overline{G}(s)\right)\bar{b}. \end{aligned}\tag{10}$$

Then, using (8) and comparing the coefficients of both sides of (10), the relation (9) directly follows.

In the following result, we show that the entries of $\overline{Z}$ can be computed from the SCFs of $\mathscr{G}(\overline{H}_o)$. Then, based on this result we propose our main result.

Lemma 8 *Denote $C_s(F_j,E_k)$ as a SCF of $\mathscr{G}(\overline{H}_o)$, which has an edge f_{1j}, for $j = 1,2,\cdots,n$ and k E-edges, for $k = 0,1,\cdots,r$. Let $w(C_s(F_j,E_k))$ be the product of weight of the edges involved in $C_s(F_j,E_k)$, by dropping term s that appears in E-edges and setting the edge weight $f_{1j} = 1$. For some fixed j and k, let $\mathscr{C}_s(j,k)$ be the set of all SCFs $C_s(F_j,E_k)$. Denote δ as the number of vertex disjoint cycles in $C_s(F_j,E_k)$. Then, the elements of matrix $\overline{Z}$ are:*

$$z_{j(k+1)} = \sum_{\mathscr{C}_s(j,k)}(-1)^{n+\delta}w\left(C_s(F_j,E_k)\right). \tag{11}$$

Proof 6 From the definition of matrix $\overline{Z}$, as defined in Theorem 7, the elements are: $z_{j(k+1)} = \sum_{l=1}^{n} d^k_{jl}b_{l1}$, where d^k_{jl} is an element of matrix $\overline{D}_k$. According to Lemma 6, d^k_{jl} can be computed from 1-connections L^k_{jl} of $\mathscr{G}(\overline{G}(s))$. Recall that each 1-connection L^k_{jl} (for $j \neq l$) contains: i) vertex disjoint cycles and a path or ii) only a path from vertex v_l to vertex v_j. Further, the 1-connection L^k_{jj} contains an isolated vertex v_j and vertex disjoint cycles. Now, consider a 1-connection L^k_{jl}, and augment it by including: i) a vertex v_{n+1}, ii) an edge from vertex v_{n+1} to v_l for $j \neq l$ (to v_j for $j = l$) with edge weight b_{l1}, and iii) an edge from vertex v_j (for both $j \neq l$ and $j = l$) to v_{n+1} with edge weight f_{1j}. Then, the resulting augmented digraph becomes a SCF $C_s(F_j,E_k)$ of $\mathscr{G}(\overline{H}_o)$. Further, it follows from the proof of Lemma 6 that the product of edge weights in $C_s(F_j,E_k)$, by dropping the term s that appears in E-edges and setting $f_{1j} = 1$, becomes a summand for $d^k_{jl}b_{l1}$. In addition, while constructing $C_s(F_j,E_k)$ from L^k_{jl}, the number of vertex disjoint cycles in $C_s(F_j,E_k)$ is increased by one. Hence, the vertex disjoint cycles δ in $C_s(F_j,E_k)$ becomes $\delta = \bar{\delta}+1$, where $\bar{\delta}$ is the number of vertex disjoint cycles in L^k_{jl}. It then follows that the sign factor for $w(C_s(F_j,E_k))$ is determined by the relation $(-1)^{n+\delta}$. For a given set of j, l and k, denote the set of SCFs $C_s(F_j,E_k)$ (constructed from L^k_{jl}) as $\mathscr{C}_s(j,k,l)$. Then, we have: $d^k_{jl}b_{l1} = \sum_{\mathscr{C}_s(j,k,l)}(-1)^{n+\delta}w(C_s(F_j,E_k))$. Now, for some fixed j and k, collect all possible L^k_{jl}'s, for $l = 1,2,\cdots,n$, and then augment them as discussed above to form $C_s(F_j,E_k)$. Note that the set formed by the collection of all such SCFs is $\mathscr{C}_s(j,k)$, and hence, $\mathscr{C}_s(j,k,l) \subset \mathscr{C}_s(j,k)$. Since $z_{j(k+1)} = \sum_{l=1}^{n} d^k_{jl}b_{l1}$, it follows that: $z_{j(k+1)} = \sum_{l=1}^{n} d^k_{jl}b_{l1} = \sum_{\mathscr{C}_s(j,k)}(-1)^{n+\delta}w(C_s(F_j,E_k))$.

Note that the free-entries of $\overline{Z}$ ($z_{ij} \neq 0$) are multivariate polynomials in variables: e_{ij}, a_{ij} and b_{i1}, and hence, they may not be (algebraically) independent to each other, which is an important criterion for a structured matrix to satisfy (see the definition in Appendix-A). To consider $\overline{Z}$ as a valid structured matrix, we need to ensure that its free-entries z_{ij} are algebraically independent. To verify this, we consider the following procedure. For $i,j = 1,2,\cdots,n$, denote the variables: e_{ij}, a_{ij} and b_{i1}, which appear in the free-entries z_{ij} of $\overline{Z}$, as $\xi_1,\xi_2,\cdots,\xi_q$. Then, z_{ij}'s are multivariate polynomials in variables $\xi_1,\xi_2,\cdots,\xi_q$ over field $\mathbb{R}$, which are denoted as: $z_1(\xi), z_2(\xi),\cdots,z_p(\xi)$, where p is the number of free-entries of $\overline{Z}$. For $k = 1,2,\cdots p$ and $l = 1,2,\cdots,q$, construct the Jacobian matrix $J(\xi)$, whose $(k,l)^{th}$ entry is: $\frac{\partial z_k}{\partial \xi_l}$ (partial derivative of z_k with respect to ξ_l). Then, it follows from the

Jacobian criterion (Ehrenborg & Rota 1993) that the multivariate polynomials $z_1(\xi), z_2(\xi), \cdots, z_p(\xi)$ are algebraically independent over $\mathbb{R}$ if and only if the Jacobian matrix $J(\xi)$ has rank p. The rank of $J(\xi)$ is p if and only if there exists a p^{th} order minor in $J(\xi)$ such that the corresponding multivariate polynomial $\psi(\xi_1, \xi_2, \cdots, \xi_q)$ is not identically equal to zero. Further, it follows from (Agrawal & Saptharishi 2009, Lemma 4.1) that if $\psi(\xi_1, \xi_2, \cdots, \xi_q)$ is not identically equal to zero, then the probability of $\psi(\tau_1, \tau_2, \cdots, \tau_q) = 0$ is close to zero, where τ_i's are the set of real numbers, which are chosen randomly with uniform distribution from a very large cardinality set $S \subset \mathbb{R}$. Hence, to verify whether the matrix $J(\xi)$ has rank p, we check the (numerical) rank of matrix $J(\tau)$, which is obtained by evaluating $J(\xi)$ at some randomly chosen real numbers τ_i's. If the (numerical) rank of $J(\tau)$ is p, then we consider $z_1(\xi), z_2(\xi), \cdots, z_p(\xi)$ to be algebraically independent, and the matrix $\overline{Z}$ to be a structured matrix.

Theorem 9 *Assume that s-rank($\overline{E}$) $= r$. Let $\overline{Z}$ be as defined in Theorem 7, and its free-entries ($z_{ij} \neq 0$) are algebraically independent. Then, the following statements hold.*

(1) There exists a feedback control of the form (3) *such that Objective 1 is achieved if and only if s-rank($\overline{Z}$) $= r+1$.*

(2) Let s-rank($\overline{Z}$) $= r+1$. Then, the minimum number of free-entries required in the feedback gain vector $\bar{f}$, by which Objective 1 can be achieved, is $r+1$.

Proof 7 Consider a numerical realization of SDS (1): $E\dot{x} = Ax + bu$, and construct: $G(s) = (sE - A)$, $\alpha(s) = \det(G(s))$ and $D(s) = \mathrm{adj}(G(s))$. Let α, which is constructed from $\alpha(s)$, be a numerical realization of $\bar{\alpha}$, where $\bar{\alpha}$ is as in (7). Further, let D_k's be the coefficient matrices of $D(s)$. Then, by defining $Z := [D_0 b\ D_1 b\ \cdots\ D_r b]$, it is clear that Z is a numerical realization of structured matrix $\overline{Z}$, as defined in Theorem 7. Corresponding to a given set of r finite poles: $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_r\}$ of $E\dot{x} = (A + bf^T)x$, construct the polynomial $\sigma(s)$. Further, construct σ from $\sigma(s)$ such that σ is a numerical realization of $\bar{\sigma}$, where $\bar{\sigma}$ is as in (7). Then, following relation is obtained from (9):

$$Z^T \bar{f} = \alpha - \sigma. \tag{12}$$

Since Z, α and σ in (12) are known quantities, the gain vector $\bar{f}$ can be obtained by solving the set of linear equations (12) with variables f_{1j}. However, (12) has a solution f (a numerical realization of $\bar{f}$) if and only if the rank of Z is $r+1$, which directly follows from the theory of linear algebra (Strang 2006). Hence, for a given set of r closed loop finite poles Λ, it is clear that a solution f of (9) can be obtained for almost all numerical realizations Z and α of $\overline{Z}$ and $\bar{\alpha}$, respectively (that is, for almost all E, A and b) if and only if s-rank($\overline{Z}$) $= r+1$. Further, s-rank($\overline{Z}$) $= r+1$ ensures that the structural degree of $\overline{\sigma}(s)$ in (9) is r. Hence, the closed loop SDS (4) is SIF, and the first statement holds.

Now, assume that s-rank($\overline{Z}$) $= r+1$. Then, by definition, one can always find a set of $r+1$ free-entries of $\overline{Z}$ such that none of them lies in same line (rows or columns). Let us denote these free-entries as $z_{\bar{p}\bar{q}}$, where $\bar{p} \in \{1, 2, \cdots, n\}$ and $\bar{q} \in \{1, 2, \cdots, r+1\}$. Now, construct a gain vector $\bar{f}^*$ from $\bar{f}$ such that the $r+1$ entries $f_{1\bar{p}}$, which are used to construct $z_{\bar{p}\bar{q}}$ (according to Lemma 8), are free-entries, and other elements f_{1j}, for $j \neq \bar{p}$, are zero. With the newly constructed gain vector $\bar{f}^*$, the relation (9) becomes:

$$\overline{Z}^T \bar{f}^* = \overline{\alpha} - \overline{\sigma}, \tag{13}$$

which can be used to achieve Objective 1 with gain vector $\bar{f}^*$. Note that for $j \neq \bar{p}$, the j^{th} row of $\overline{Z}$ does not play any role in solving (13), since $f_{1j} = 0$ in $\bar{f}^*$. Hence, one can set the j^{th} row of $\overline{Z}$ to zero in (13) without affecting its solution. With the similar argument, if more than $n - (r+1)$ entries of $\bar{f}$ are zero, then the corresponding rows of $\overline{Z}$ can be set to zero, since they have no contributions in solving (13). As a consequence, however, the structural rank of modified $\overline{Z}$ will become less than $r+1$. Hence, Objective 1 can not be achieved, according to the first statement, which implies the second statement holds. There is a possibility that one can find more than one set of $r+1$ free-entries $z_{\bar{p}\bar{q}}$ of $\overline{Z}$. Hence, the choice of $\bar{f}^*$ is not unique, which will change according to the choice of $z_{\bar{p}\bar{q}}$. However, only one of such sets is required to achieve Objective 1.

4 Relevant algorithms to address Problem 2

According to Theorem 9, the matrix $\overline{Z}$ plays an important role in addressing Problem 2. Moreover, according to Lemma 8, the entries of $\overline{Z}$ are determined by the SCFs of digraph $\mathscr{G}(\overline{H}_0)$. Hence, we now first present an algorithm to compute the SCFs of $\mathscr{G}(\overline{H}_0)$, and the vertex disjoint cycles in a SCF. For this, we use Proposition 11 and Proposition 12, presented in Appendix-B.

Stepwise explanations of Algorithm 1: *Steps 1-6*: To implement the algorithm in a mathematical software, we assign some randomly chosen non-zero real numbers to the free-entries of $\overline{E}$, $\overline{A}$ and $\bar{b}$, so that their zero/free-entry patterns can be preserved. Further, since the exact structure of the feedback gain vector $\bar{f}$ is not known a priori, it is assumed that all of its elements f_{1j} are free-entries, and hence, they are assigned with some randomly chosen non-zero real numbers (Step 1). In Step 2, $\overline{H}_0$ and $\mathscr{G}(\overline{H}_0)$ are constructed as discussed in Section 3.1. In Step 3, the number of vertices in $\mathscr{G}(\overline{H}_0)$ is set to $\nu = n+1$, where n is the number of states in (1). Further, the number of edges in $\mathscr{G}(\overline{H}_0)$ is counted and set it to m. This step has complexity $\mathscr{O}(m)$. Note that $\mathscr{G}(\overline{H}_0)$ may contain parallel edges (if e_{ij} and a_{ij} are free-entries of $\overline{E}$ and $\overline{A}$). Hence, to distinguish these two edges, a vector ω_t is defined in Step 4 to assign a unique identification number (UIN) to each edge of $\mathscr{G}(\overline{H}_0)$. For $i = 1, 2, \cdots, m$, the i^{th} element of ω_t is defined as: $\omega_t(i) = n + i + \kappa$, where κ is any positive number, which without loss of generality, is chosen as 2. This step has complexity $\mathscr{O}(m)$. The incidence matrix I_g of $\mathscr{G}(\overline{H}_0)$, without considering its self-loops, is con-

Algorithm 1 Algorithm for computing SCFs of $\mathscr{G}(\overline{H}_0)$

Input: The structured matrices $\overline{E}$, $\overline{A}$, $\bar{b}$.
Output: The matrices: Θ_{sw} and Θ_{tw}.

1: Assign some randomly chosen non-zero real numbers to the free-entries of $\overline{E}$, $\overline{A}$ and $\bar{b}$, and to all the entries of $\bar{f}$.
2: Construct $\overline{H}_0$ and $\mathscr{G}(\overline{H}_0)$ as discussed in Section 3.1.
3: Set $\nu = n+1$ to be the number of vertices, and m to be the number of edges in $\mathscr{G}(\overline{H}_0)$.
4: Define $\omega_t = [\omega_{t_1}\ \omega_{t_2}\ \cdots\ \omega_{t_m}]$, where $\omega_{t_i} = \nu + i + \kappa$.
5: Construct the incidence matrix I_g of $\mathscr{G}(\overline{H}_0)$ by removing its self-loops. Define the matrices: I_{ga} as in Proposition 12 and $I_{gw} := \begin{bmatrix} I_{ga} \\ \omega_t \end{bmatrix}$.
6: Define $\eta = [1\ 2\ \cdots\ m]$. Construct a matrix C_{mb} of size $p \times \nu$, where $p = \frac{m!}{(m-\nu)!\nu!}$, by considering all possible combinations of the elements of η, taken ν at a time.
7: **for** $i = 1$ to p **do**
8: **If** matrix $I_{ga}(:, C_{mb}(i,:))$ satisfies condition-2 and condition-3 of Proposition 11, **then:** i) store it in S, and the matrix $I_{gw}(:, C_{mb}(i,:))$ in S_w, ii) identify the edges from the columns of S, and store the initial and final vertices of those edges in vectors c_s and c_t, respectively, iii) store $[c_s\ S_w(\nu,:)]$ and $[c_t\ S_w(\nu,:)]$ in the i^{th} row of T_{sw} and T_{tw}, respectively.
9: **else if** i) there is at least one zero column in matrix $I_{ga}(:, C_{mb}(i,:))$, ii) $I_{ga}(:, C_{mb}(i,:))$ satisfies condition-2 and condition-3 of Proposition 12 and iii) there are self-loops at the vertices of $\mathscr{G}(\overline{H}_0)$ corresponding to the zero rows of $I_{ga}(:, C_{mb}(i,:))$, **then:** i) store $I_{ga}(:, C_{mb}(i,:))$ in S_l, and $I_{gw}(:, C_{mb}(i,:))$ in S_{lw}, ii) identify the edges from the non-zero columns of S_l, and store the initial and final vertices of those edges in c_{sl} and c_{tl}, respectively, iii) construct $l_{sl} = [c_{sl}\ s_{pl}]$ and $l_{tl} = [c_{tl}\ s_{pl}]$, and iv) store $[l_{sl}\ S_{wl}(\nu,:)]$ and $[l_{tl}\ S_{wl}(\nu,:)]$ in the i^{th} row of T_{swl} and T_{twl}, respectively. The quantity s_{pl} refers to the position of zero rows in S_l.
10: **end if**
11: **end if**
12: **end for**
13: Set $\Theta_{sw} = \begin{bmatrix} T_{sw} \\ T_{swl} \end{bmatrix}$ and $\Theta_{tw} = \begin{bmatrix} T_{tw} \\ T_{twl} \end{bmatrix}$.

structed in Step 5. Then, I_{ga} is constructed as in Proposition 12, by including the self-loops. Note that each column of I_{ga} corresponds to an edge of $\mathscr{G}(\overline{H}_0)$. Hence, the matrix I_{gw} is defined, by stacking ω_t with I_{ga}, so that the associated edges of $\mathscr{G}(\overline{H}_0)$ will get assigned with its UIN. To compute all the possible SCFs of $\mathscr{G}(\overline{H}_0)$, we take all possible combinations of $\nu = n+1$ columns of I_{ga}, and then verify if the conditions of Proposition 11 and Proposition 12 are satisfied. For this, a matrix C_{mb} is constructed in Step 6, whose rows contains all possible combinations of the numbers from 1 to m, taken ν at a time. Since Step 6 has complexity $\mathscr{O}(\min(m^{\nu}, m^{m-\nu}))$, the overall complexity of Steps 1-6 is $\mathscr{O}(\min(m^{\nu}, m^{m-\nu}))$.

Step 8: In this step, we obtain all possible SCFs of $\mathscr{G}(\overline{H}_0)$, where no self-loops appear. The matrices $I_{ga}(:, C_{mb}(i,:))$ and $I_{gw}(:, C_{mb}(i,:))$ are obtained by considering a particular combination (determined by i^{th} row of C_{mb}) of the columns of matrices I_{ga} and I_{gw}, respectively. The matrix $I_{gw}(:, C_{mb}(i,:))$ is constructed so as to keep track of the UINs of the edges, associated with the columns of $I_{ga}(:, C_{mb}(i,:))$. The matrix $I_{ga}(:, C_{mb}(i,:))$, which satisfies the conditions of Proposition 11, is stored in S, and the corresponding matrix $I_{gw}(:, C_{mb}(i,:))$ is stored in S_w. According to Proposition 11, the edges, determined by the columns of S, form a SCF of $\mathscr{G}(\overline{H}_0)$. To verify the conditions of Proposition 11, $\mathscr{O}(n^2)$ additions and $\mathscr{O}(n^2)$ comparisons are required. Depending upon the positions of 1 and -1 in a column of S, the edges of $\mathscr{G}(\overline{H}_0)$ are identified, and the initial and final vertices of these edges are stored in vectors c_s and c_t, respectively. Since the UINs of the edges, associated with the constructed SCF, are available in the last row of S_w, the i^{th} row of matrices T_{sw} and T_{tw}, are constructed as: $T_{sw}(i,:) = [c_s\ \ S_w(\nu,:)]$ and $T_{tw}(i,:) = [c_t\ \ S_w(\nu,:)]$, respectively. The overall complexity of this step, at each iteration, is $\mathscr{O}(n^2)$.

Step 9: In this step, we obtain all possible SCFs of $\mathscr{G}(\overline{H}_0)$, which are having at least one self-loop. If $I_{ga}(:, C_{mb}(i,:))$ does not satisfy the conditions of Proposition 11, then at Step 9, it is verified if the conditions of Proposition 12 are satisfied. If $I_{ga}(:, C_{mb}(i,:))$ satisfies the conditions of Proposition 12, and there are self-loops at the vertices of $\mathscr{G}(\overline{H}_0)$, corresponding to the zero rows of $I_{ga}(:, C_{mb}(i,:))$, then $I_{ga}(:, C_{mb}(i,:))$ and $I_{gw}(:, C_{mb}(i,:))$ are stored in S_l and S_{lw}, respectively. It then follows from Proposition 12 that the edges characterized by the non-zero columns and zero rows of S_l will form a SCF (the zero rows correspond to the self-loops). The edges, corresponding to the non-zero columns of S_l, are identified, and the initial and final vertices of these edges are stored in c_{sl} and c_{tl}, respectively. Further, the position of zero rows in S_l are identified, and stored them in s_{pl}. Then, $l_{sl} = [c_{sl}\ s_{pl}]$ and $l_{tl} = [c_{tl}\ s_{pl}]$ are constructed. Finally, using the UINs of the edges, the i^{th} row of matrices: T_{swl} and T_{twl}, are updated as follows: $T_{swl}(i,:) = [l_{sl}\ S_{wl}(\nu,:)]$ and $T_{twl}(i,:) = [l_{tl}\ S_{wl}(\nu,:)]$. Since the complexity of this step, at each iteration, is $\mathscr{O}(n^2)$, the complexity for Step 7 to Step 12 is: $\mathscr{O}(n^2 p)$, where $p = \binom{m}{\nu}$. Hence, the overall complexity of Algorithm 1 is: $\mathscr{O}\left(\min(m^{\nu}, m^{m-\nu}) + n^2 p\right)$. Finally, at Step 13, $\Theta_{sw} = \begin{bmatrix} T_{sw} \\ T_{swl} \end{bmatrix}$ and $\Theta_{tw} = \begin{bmatrix} T_{tw} \\ T_{twl} \end{bmatrix}$ are constructed. Note that $\Theta_{sw}(i, 1:\nu)$ and $\Theta_{tw}(i, 1:\nu)$ correspond to the initial and final vertices of the edges, respectively, involved in the i^{th} SCF of $\mathscr{G}(\overline{H}_0)$. The entries in $\Theta_{sw}(i, \nu+1:2\nu)$ and $\Theta_{tw}(i, \nu+1:2\nu)$ are equal, and they represent the UINs of the edges involved in i^{th} SCF.

Explanation for Algorithm 2: Two vectors: v_{cs} and v_{ct} are given as inputs to the algorithm. In Step 1, the number of self-loops, denoted as n_{sl}, are computed by verifying whether $v_{cs}(i) = v_{ct}(i)$, for $i = 1, 2, \cdots, \nu$. In Step 2, $\bar{v}_{cs}$ and $\bar{v}_{ct}$ are constructed by removing the self-loop edges from v_{cs} and v_{ct}. Using $\bar{v}_{cs}$ and $\bar{v}_{ct}$, a new matrix V_l is defined, where the j^{th} column of V_l represents an edge of a cycle in the given SCF. In Step 5, the first column of V_l is considered as

Algorithm 2 Algorithm for computing vertex disjoint cycles in a SCF.

Input: The vectors: v_{cs} and v_{ct}, where the elements of v_{cs} and v_{ct} are the indices of initial and final vertices of the edges in a SCF, respectively.

Output: The number of vertex disjoint cycles δ in the SCF.

1: Compute the number of self-loops (n_{sl}) in the SCF, by verifying if $v_{cs}(i) = v_{ct}(i)$, for $i = 1, 2, \cdots, v = n+1$.
2: Construct $\bar{v}_{cs}$ and $\bar{v}_{ct}$ from v_{cs} and v_{ct}, respectively, by removing the self-loop edges. Set $V_l = \begin{bmatrix} \bar{v}_{cs} \\ \bar{v}_{ct} \end{bmatrix}$.
3: Set $l_p = 1$, $n_l = 0$ and $\bar{l} =$ number of columns in V_l.
4: **for** $i = 1$ to $\bar{l}$ **do**
5:     Starting from $V_l(:, l_p)$, find a set of edges from V_l such that the initial vertex of succeeding edge is equal to the final vertex of preceding edge. Continue this process till $(V_l(1,1))$ becomes the final vertex of an edge in V_l. Store the index of all these columns of V_l in x_p.
6:     Set $n_l = n_l + 1$.
7:     Remove the columns of V_l, indexed by x_p. Set $\bar{l} =$ number of remaining columns in V_l, and $l_p = 1$.
8:     **if** V_l is empty, **then**
9:         $\delta = n_l + n_{sl}$.
10:         **stop**.
11:     **end if**
12: **end for**

first edge, and then, a set of edges are searched in V_l such that the initial vertex of succeeding edge is equal to the final vertex of preceding edge. This process is continued until the initial vertex of first edge of this set becomes the final vertex of an edge in V_l. Then, according to the definition, these set of edges will form a cycle. Once such a set of edges are found out from V_l, the number of cycles n_l in the given SCF is updated by $n_l = n_l + 1$ in Step 6. Then, the set of edges (indexed by x_p), corresponding to the obtained cycle, are removed from V_l in Step 7. This process is continued until V_l becomes empty. By this time, we have counted all the cycles n_l (excluding self-loops) in the SCF. Hence, the total number of vertex disjoint cycles in the given SCF is $\delta = n_l + n_{sl}$, where n_{sl} is the number of self-loops in the SCF (Step 9). All the steps in this algorithm have linear complexity, except Step 5, which is of $\mathscr{O}(n^2)$ complexity. Since a cycle contains at least two edges, to make V_l empty, at most $\frac{v}{2}$ (or the largest integer less than or equal to $\frac{v}{2}$) number of iterations are required. Hence, the overall complexity of this algorithm is $\mathscr{O}(n^3)$.

Recall from Theorem 9 that Problem 2 is solvable if and only if s-rank$(\overline{Z}) = r+1$. Furthermore, it follows from the proof of Theorem 9 that a minimum free-entries gain vector $\bar{f}^*$ can be obtained by finding a set of $r+1$ entries $z_{\bar{p}\bar{q}}$ of $\overline{Z}$, which do not lie in a same line (rows or columns). To determine these quantities we associate a bipartite graph (see Appendix-A), denoted as $\mathscr{G}_b(\overline{Z})$, with the structured matrix $\overline{Z}$. Then, it follows from (Murota 1987*b*, Reinschke & Wiedemann 1997) that s-rank$(\overline{Z})$ is equal to the number of edges appear in a *maximum matching* of $\mathscr{G}_b(\overline{Z})$. Hence, to determine s-rank$(\overline{Z})$, we compute a maximum matching of $\mathscr{G}_b(\overline{Z})$, which can be solved in polynomial time using existing algorithms, such as Hopcroft & Karp (1973) algorithm (complexity $\mathscr{O}(h\sqrt{n})$, where h is the number of free-entries in $\overline{Z}$). Since the edges in a maximum matching of $\mathscr{G}_b(\overline{Z})$ do not share a common vertex, it is easy to notice that the weights of the edges in a maximum matching of $\mathscr{G}_b(\overline{Z})$ are the entries $z_{\bar{p}\bar{q}}$ of $\overline{Z}$, which do not lie in a same line (rows or columns). Hence, if s-rank$(\overline{Z}) = r+1$, the weights of $r+1$ edges in a maximum matching of $\mathscr{G}_b(\overline{Z})$ are the $r+1$ entries $z_{\bar{p}\bar{q}}$ of $\overline{Z}$, which do not lie in a same line (rows or columns).

5 Demonstrative Example

Example 1 Consider a SDS of the form (1), where the system matrices: $\overline{E}$, $\overline{A}$ and $\bar{b}$ have the following structures: $\overline{E} = \begin{bmatrix} 0&0&0&0 \\ 0&0&\star&0 \\ 0&0&\star&0 \\ \star&0&0&0 \end{bmatrix}$, $\overline{A} = \begin{bmatrix} 0&\star&\star&0 \\ \star&0&0&0 \\ 0&0&\star&0 \\ \star&0&0&\star \end{bmatrix}$, and $\bar{b} = \begin{bmatrix} 0 \\ 0 \\ \star \\ 0 \end{bmatrix}$. It can be noticed from $\mathscr{G}(\overline{G}(s))$, depicted in Fig. 1, that there are no SCFs having 2-E edges. Hence, it follows from (Reinschke 1994, Theorem 1) that the structural degree of $\bar{\alpha}(s)$ is strictly less than 2. Since s-rank$(\overline{E}) = 2$, according to Definition 1, the open loop system is not SIF. Considering $\bar{f} = [\star\ \star\ \star\ \star]^T$, the structured matrix $\overline{H}_o$ is constructed as in (6), and the associated digraph $\mathscr{G}(\overline{H}_o)$ is shown in Fig. 1.

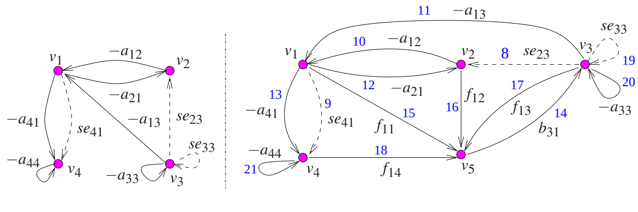

Fig. 1. Digraphs $\mathscr{G}(\overline{G}(s))$ (left side) and $\mathscr{G}(\overline{H}_o)$ (right side). The UINs assigned to the edges of $\mathscr{G}(\overline{H}_o)$ are marked in blue color.

To obtain the SCFs of $\mathscr{G}(\overline{H}_o)$, we implement Algorithm 1 in MATLAB (version: R2020b), and computed: $\Theta_{sw} = \begin{bmatrix} 3&1&2&5&4&8&9&10&14&18 \\ 3&2&1&5&4&8&10&13&14&18 \\ 3&2&5&1&4&8&10&14&15&21 \\ 2&1&5&3&4&10&12&14&17&21 \\ 3&1&5&2&4&11&12&14&16&21 \end{bmatrix}$, $\Theta_{tw} = \begin{bmatrix} 2&4&1&3&5&8&9&10&14&18 \\ 2&1&4&3&5&8&10&13&14&18 \\ 2&1&3&5&4&8&10&14&15&21 \\ 1&2&3&5&4&10&12&14&17&21 \\ 1&2&3&5&4&11&12&14&16&21 \end{bmatrix}$. For $i = 1, 2, \cdots, 5$, $\Theta_{sw}(i, 1:5)$ and $\Theta_{tw}(i, 1:5)$ correspond to the indices of initial and final vertices of the edges, respectively, involved in the i^{th} SCF of $\mathscr{G}(\overline{H}_0)$. The entries in $\Theta_{sw}(i, 6:10)$ and $\Theta_{tw}(i, 6:10)$ are equal, and they represent the UINs of the edges involved in i^{th} SCF, as depicted in Fig. 2.

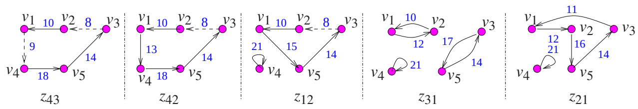

Fig. 2. SCFs of $\mathscr{G}(\overline{H}_o)$, where the blue colored numbers are the UINs of the edges.

Since there exists a SCF having 2 E-edges, it follows from Theorem 5 that the closed loop SDS (4) can be made SIF, and the associated gain vector is: $\bar{f} = [0\ 0\ 0\ f_{14}]^T$. According to Lemma 8, we obtained: $\overline{Z} = \begin{bmatrix} 0&\star&\star&0 \\ \star&0&0&\star \\ 0&0&0&\star \end{bmatrix}^T$, where the free-entries are: $z_{12} = e_{23}a_{12}a_{44}b_{31}$, $z_{21} = -a_{13}a_{21}a_{44}b_{31}$, $z_{31} =$

$a_{12}a_{21}a_{44}b_{31}$, $z_{42} = -e_{23}a_{12}a_{41}b_{31}$ and $z_{43} = e_{23}e_{41}a_{12}b_{31}$. Note that the free-entries z_{ij} are the multivariate polynomials in variables: $\xi_1 = e_{23}$, $\xi_2 = e_{41}$, $\xi_3 = a_{12}$, $\xi_4 = a_{13}$, $\xi_5 = a_{21}$, $\xi_6 = a_{41}$, $\xi_7 = a_{44}$ and $\xi_8 = b_{31}$. According to the procedure stated in Section 3, the Jacobian matrix $J(\xi)$ of size 5×8 is evaluated at some randomly chosen real numbers, and verified that its (numerical) rank is 5. Hence, the free-entries z_{ij} are algebraically independent, and $\overline{Z}$ is a structured matrix. To compute s-rank$(\overline{Z})$, we associate the bipartite graph $\mathscr{G}_b(\overline{Z})$, as shown in Fig. 3, with $\overline{Z}$. Then, a maximum matching of $\mathscr{G}_b(\overline{Z})$ is obtained, which is shown in Fig. 3, using the command *matching*, available in *SageMath* (SageMath n.d.). From the computed maximum matching, we find that the following free-entries z_{12}, z_{21} and z_{43} of $\overline{Z}$ do not lie in a same line (rows or columns). It then follows from the proof of Theorem 9 and Lemma 8 that $\bar{f}^* = [\star\ \star\ 0\ \star]^T$. Notice that the measurement of state, corresponding to the zero entry of $\bar{f}^*$, is not required. Hence, the associated state measurement sensor can be eliminated in the implementation, which helps in reducing the cost. For the choice of following randomly chosen free-entries: $e_{23} = 0.9058$, $e_{33} = 0.1270$, $e_{41} = 0.8147$, $a_{12} = 0.1538$, $a_{13} = 0.8989$, $a_{21} = 0.0957$, $a_{33} = 0.7120$, $a_{41} = 0.4231$, $a_{44} = 0.2389$ and $b_{31} = 0.9575$, the open loop finite pole is 5.6063. By choosing the closed loop finite poles as: $-1 \pm 2i$, we obtained: $Z = \begin{bmatrix} 0 & -0.0197 & 0.0034 & 0 \\ 0.0319 & 0 & 0 & -0.0564 \\ 0 & 0 & 0 & 0.1087 \end{bmatrix}^T$, which is a numerical realization of $\overline{Z}$. Further, we obtained: $f^* = [-79.0433\ \ 254.2197\ \ 0\ \ -9.2018]^T$, which is a numerical realization of $\bar{f}^*$.

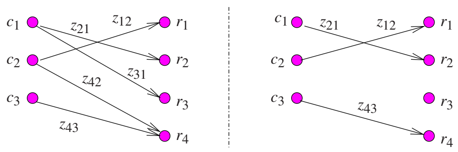

Fig. 3. The bipartite graph $\mathscr{G}_b(\overline{Z})$ (left side) of $\overline{Z}$, and a maximum matching of $\mathscr{G}_b(\overline{Z})$ (right side).

6 Concluding Remarks

In this work, we proposed a methodology and associated algorithms to determine the existence of a static state feedback control for a SDS such that almost all of its numerical realizations can be made impulse free and place their finite poles at any desired locations in the complex plane. Further, the computation of a minimum free-entry pattern gain vector is proposed. To address the problems, graph theoretic method is used. The developed algorithms are verified with a numerical example, where satisfactory results are obtained. The proposed approach has limitation in applying to a large-scale system due to the computational cost associated with: i) verifying the algebraic independence of entries of the matrix $\overline{Z}$, and ii) computation of all the SCFs of a digraph. However, with numerical experimentation, it has been observed that the developed algorithms perform satisfactorily with small-scale systems. The advantages of the proposed work are as follows. The developed approach and algorithms are not specific to a particular system, rather they are applicable to a class of DSs (having same structure). The computation of a minimum free-entry pattern gain vector helps in determining the state measurement sensors that can be eliminated so as to reduce the implementation cost. Finally, the proposed framework helps in identifying a set of critical feedback paths (associated with the free-entries of a minimum free-entry pattern gain vector), which need to be secured to ensure satisfactory closed-loop performance against cyber-attacks. It would be interesting to extend the proposed work for multi-input SDS, which however, is left for further investigation, since the proposed results can not directly be used for multi-input systems.

Appendix - A

Structured matrix and related definitions: A matrix $\overline{M}$ is said to be a *structured matrix* if its entries are either fixed at 0 or have indeterminate values (free-entries), which are assumed to be *independent* to each other (Murota 1987*b*, Reinschke 1988). The free elements of $\overline{M}$ are denoted as $\star$. When the free-entries of $\overline{M}$ are assigned with some randomly chosen real numbers, the resulting matrix M is referred to as a numerical realization of $\overline{M}$. Since the free-entries of $\overline{M}$ can take any real numbers, including zero, there is a possibility that a property under investigation, such as rank and determinant of $\overline{M}$, does not hold for *all* M. We say that a property holds *structurally* (generically) for $\overline{M}$, if it holds for *almost all* M (the values of parameters for which the property may not hold belong to a set of Lebesgue-measure zero (Reinschke 1988, Dion et al. 2003)). The *structural rank* of $\overline{M}$ is the maximum number of free-entries of $\overline{M}$ such that none of them lies in a same line (rows or column). We say that $\overline{M}$ is *structurally singular* (*structurally non-singular*), if $\det(\overline{M}) \equiv 0$ ($\det(\overline{M}) \not\equiv 0$), irrespective to the entries of $\overline{M}$.

Digraph representation of a structured matrix: Consider a structured matrix $\overline{M}$ of size $n \times m$, and define $\mu := \max(n,m)$, where max refers to the maximum. Corresponding to $\overline{M}$, we associate a digraph $\mathscr{G}(\overline{M})$ on the set of μ vertices: $\mathscr{V} := \{v_1, v_2, \cdots, v_\mu\}$, where a vertex v_i represents a column or row of $\overline{M}$, whereas an edge, denoted as (v_j, v_i), having direction from v_j (initial vertex) to v_i (final vertex), represents a free-entry m_{ij} of $\overline{M}$. A *path*, from vertex v_k to v_l, in $\mathscr{G}(\overline{M})$ is a sequence of vertices, where the following conditions hold: i) no edge enters at v_k (initial vertex of the path), ii) no edge leaves from v_l (final vertex of the path) and iii) only one edge enters and one edge leaves from the intermediate vertices. A closed path, whose initial and final vertices are same, is referred to as *cycle*. A cycle with one edge is referred to as *self-loop*. A set of vertex disjoint cycles (including self-loops) is referred to as *cycle family*. If the cycle family touches all the vertices in $\mathscr{G}(\overline{M})$, then it is referred to as *spanning cycle family* of $\mathscr{G}(\overline{M})$ (Reinschke 1988). Denote $\mathscr{E}$ as the set of edges of $\mathscr{G}(\overline{M})$. Then, a digraph having vertex set $\mathscr{W} \subseteq \mathscr{V}$ and edge set $\mathscr{F} \subseteq \mathscr{E}$ is referred to as *subdigraph* of $\mathscr{G}(\overline{M})$. When $\mathscr{W} = \mathscr{V}$, the subdigraph is referred to as *spanning subdigraph* of $\mathscr{G}(\overline{M})$.

For a digraph $\mathscr{G}(\overline{M})$, a 1-*connection*, from vertex v_j to vertex v_i, denoted as L_{ij}, is a *spanning subdigraph* of $\mathscr{G}(\overline{M})$ having following properties (Brualdi & Cvetkovic 2009): 1) *for* $i \neq j$: i) at vertex v_i, exactly one edge enters and no edge leaves, ii) at vertex v_j, exactly one edge leaves, but no edge enters, and iii) for vertex $v_k \neq v_i, v_j$, exactly one edge enters and exactly one edge leaves, 2) *for* $i = j$: i) no edges enter or leave at vertex v_i, and ii) for vertex $v_k \neq v_i$, exactly one edge enters and exactly one edge leaves.

Structural rank computation: For a given structured matrix $\overline{M}$, of size $n \times m$, we can associate a bipartite graph $\mathscr{G}_b(\overline{M})$ on the set of $n+m$ vertices, where n vertices: $r_1, r_2, \cdots, r_n$ correspond to n rows of $\overline{M}$, and m vertices: $c_1, c_2, \cdots, c_m$ correspond to m columns of $\overline{M}$. In $\mathscr{G}_b(\overline{M})$, there is an edge from vertex c_j to vertex r_i with weight m_{ij}, if m_{ij} is a free-entry of $\overline{M}$. A *matching* in $\mathscr{G}_b(\overline{M})$ refers to a subset of edges such that none of the two edges in the subset have a common vertex. A matching in $\mathscr{G}_b(\overline{M})$, having maximum number of edges, is referred to as *maximum matching*.

Proposition 10 *Consider a structured matrix $\overline{M}$ with its digraph $\mathscr{G}(\overline{M})$. Let $\mathscr{L}(\overline{M})$ be a spanning subdigraph of $\mathscr{G}(\overline{M})$ such that at each vertex of $\mathscr{L}(\overline{M})$ at most one edge either enters or leaves. Denote $|E_L|$ as the number of edges in $\mathscr{L}(\overline{M})$. Then, s-rank($\overline{M}$) $= \max_{\mathscr{L}(\overline{M})} |E_L|$.*

Proof 8 Note that one can obtain the digraph $\mathscr{G}(\overline{M})$ from its corresponding bipartite graph $\mathscr{G}_b(\overline{M})$ by combining the vertices c_i and r_i of $\mathscr{G}_b(\overline{M})$ and forming a single new vertex v_i for $\mathscr{G}(\overline{M})$. With this, an edge between the vertices c_i and r_i in $\mathscr{G}_b(\overline{M})$ becomes a self-loop in $\mathscr{G}(\overline{M})$ at vertex v_i. Similarly, an edge between the vertices c_i and r_j in $\mathscr{G}_b(\overline{M})$ becomes an edge in $\mathscr{G}(\overline{M})$ between the vertices v_i and v_j. It is assumed that at each vertex v_k of $\mathscr{L}(\overline{M})$, at most one edge either enters or leaves. Since the vertex v_k in $\mathscr{L}(\overline{M})$ becomes vertices r_k and c_k in the corresponding bipartite graph $\mathscr{L}_b(\overline{M})$, no more than one edge can leave from vertex c_k in $\mathscr{L}_b(\overline{M})$ (if so, then there must be two edges leaving from vertex v_k, which is not the case in $\mathscr{L}(\overline{M})$). Similarly, no more than one edge can enter at vertex r_k in $\mathscr{L}_b(\overline{M})$. Hence, the edges that appear in $\mathscr{L}(\overline{M})$, form a matching in $\mathscr{G}_b(\overline{M})$. Therefore, $\mathscr{L}(\overline{M})$, which contains maximum number of edges, will form a maximum matching in $\mathscr{G}_b(\overline{M})$. It is shown in Murota (1987*b*), Reinschke & Wiedemann (1997) that s-rank($\overline{M}$) is equal to the number of edges appear in a maximum matching of $\mathscr{G}_b(\overline{M})$, and hence, the result holds.

Appendix - B

Incidence matrix of a digraph: Consider a digraph $\mathscr{G}$ on the set of n vertices and m edges with no self-loops. The $(i,j)^{th}$ element of *incidence matrix* I_g (size $n \times m$) of $\mathscr{G}$ is: i) -1, if j^{th} edge leaves from v_i, ii) 1, if j^{th} edge terminates at v_i and iii) 0 otherwise. Each row and column of I_g correspond to a vertex and an edge of $\mathscr{G}$, respectively. Assume that a matrix $\widetilde{I}_g$ of size $n \times m$ is given, whose columns have at most two non-zero elements: 1 and -1. Then, a digraph $\widetilde{\mathscr{G}}$, on n vertices and m edges, can be constructed from $\widetilde{I}_g$ as follows: assign an edge from vertex v_k to v_l in $\mathscr{G}$, if j^{th} column, of $\widetilde{I}_g$ has non-zero elements: $\widetilde{I}_g(k,j) = -1$ and $\widetilde{I}_g(l,j) = 1$. We use these definition and construction procedure in the following results.

Proposition 11 *Consider a digraph $\mathscr{G}$ on the set of n vertices and m edges with no self-loops and $m \geq n$. Let I_g be its incidence matrix, and a matrix $\overline{I}_g$ be constructed by stacking some or all columns of I_g. Let $\mathscr{C}_s$ be a digraph constructed from $\overline{I}_g$. Then, $\mathscr{C}_s$ becomes a SCF of $\mathscr{G}$, if the following conditions hold: 1) the number of columns of $\overline{I}_g$ is equal to n, 2) there are no rows in $\overline{I}_g$, whose all elements are zero and 3) the addition of all columns is equal to a zero vector.*

Proof 9 Since condition-1 holds, the number of vertices in $\mathscr{C}_s$ is equal to the number of edges. Condition-2 implies that none of the vertices in $\mathscr{C}_s$ is isolated, that is, at each vertex there must be at least one edge either enter or leave. Further, since condition-3 holds, at each vertex in $\mathscr{C}_s$, equal number of edges enter and leave. Since the number of vertices in $\mathscr{C}_s$ is equal to the number of edges, it is clear that at each vertex of $\mathscr{C}_s$, only one edge can enter and one edge can leave. Hence, by definition, $\mathscr{C}_s$ is a SCF of $\mathscr{G}$.

Proposition 12 *Consider a digraph $\mathscr{G}$ on the set of n vertices and m edges with $m \geq n$. Let I_g be the incidence matrix of $\mathscr{G}$ without considering its self-loops. Define a matrix $I_{ga} := [I_g \quad \mathbf{0}_{n\times l}]$, where l is the number of self-loops in $\mathscr{G}$. Let $\overline{I}_{ga}$ be a matrix, which is constructed by stacking some or all columns of I_{ga}. Assume that $\overline{I}_{ga}$ satisfies the following conditions: 1) the number of columns in $\overline{I}_{ga}$ is equal to n, 2) there are equal number of zero rows and columns, and 3) the addition of all columns is equal to a zero vector. Let $\mathscr{C}_s$ be the digraph constructed from $\overline{I}_{ga}$ as follows: i) the edges, corresponding to the non-zero columns of $\overline{I}_{ga}$, are assigned to the vertices of $\mathscr{C}_s$ according to the procedure given in the first paragraph of this Appendix, and ii) self-loops are assigned to the vertices corresponding to the zero rows of $\overline{I}_{ga}$. Then, $\mathscr{C}_s$ becomes a SCF of $\mathscr{G}$, if there exist self-loops at the vertices of $\mathscr{G}$ corresponding to the zero rows of $\overline{I}_{ga}$.*

Proof 10 Let the number of non-zero rows in $\overline{I}_{ga}$ be n_r. Since condition-2 holds, the number of non-zero columns in $\overline{I}_{ga}$ also be n_r. Without loss of generality, assume that the first n_r rows and columns of $\overline{I}_{ga}$ are non-zero (possibly it needs relabeling of vertices and edges in $\mathscr{G}$), and denote that submatrix as $\overline{I}_{sga}$. Note that $\overline{I}_{sga}$ satisfies all the conditions of Proposition 11. Hence, it follows from Proposition 11 that the subdiagraph of $\mathscr{C}_s$, corresponding to the vertices associated with $\overline{I}_{sga}$, is a SCF on n_r vertices. Since the remaining $n - n_r$ vertices of $\mathscr{C}_s$ are isolated (corresponding to the zero rows), it is apparent that $\mathscr{C}_s$ becomes a SCF of $\mathscr{G}$, if there are self-loops at the vertices of $\mathscr{G}$ corresponding to the $n - n_r$ zero rows of $\overline{I}_{ga}$.

Acknowledgment

The authors would like to thank the Editors and the anonymous reviewers for their helpful suggestions on improving the presentation of this article. The first author would like to acknowledge the helpful discussion with Dr. D. U. Patil.

References

Agrawal, M. & Saptharishi, R. (2009), Classifying polynomials and identity testing, *in* 'Current Trends in Science, Indian Academy of Sciences', pp. 149–162.

Blanchini, F. & Giordano, G. (2021), 'Structural analysis in biology: A control-theoretic approach', *Automatica* **126(109376)**, 1–23.

Boukhobza, T., Hamelin, F. & Sauter, D. (2006), 'Observability of structured linear systems in descriptor form : A graph-theoretic approach', *Automatica* **42**(4), 629–635.

Brualdi, R. A. & Cvetkovic, D. (2009), *A Combinatorial Approach to Matrix Theory and Its Applications*, CRC Press, Taylor & Francis Group, Boca Raton, NW.

Chowdhury, N. R., Belikov, J., Baimel, D. & Levron, Y. (2020), 'Observer-based detection and identification of sensor attacks in networked CPSs', *Automatica* **121(109166)**, 1–13.

Clark, A., Alomair, B., Bushnell, L. & Poovendran, R. (2017), 'Input selection for performance and controllability of structured linear descriptor systems', *SIAM Journal on Control and Optimization* **55**(1), 457–485.

Dai, L. (1989), *Singular Control Systems*, Springer-Verlag, Berlin, Germany.

Datta, S. (2017), 'Feedback controller norm optimization for linear time invariant descriptor systems with pole region constraint', *IEEE Transactions on Automatic Control* **62**(6), 2794–2806.

Dion, J. M., Commault, C. & Woude, J. V. D. (2003), 'Generic properties and control of linear structured systems: a survey', *Automatica* **39**, 1125–1144.

Duan, G. R. (2010), *Analysis and Design of Descriptor Linear Systems*, Springer-Verlag, New York, NY.

Ehrenborg, R. & Rota, G. C. (1993), 'Apolarity and canonical forms for homogeneous polynomials', *Europ. J. Combinatorics* **14**, 157–181.

Hopcroft, J. E. & Karp, R. M. (1973), 'An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs', *SIAM Journal on Computing* **2**(4), 225–231.

Lin, C. T. (1974), 'Structural controllability', *IEEE Transactions on Automatic Control* **19**(3), 201–208.

Liu, P., Zhang, Q., Yang, X. & Yang, L. (2008), 'Passivity and optimal control of descriptor biological complex systems', *IEEE Transactions on Automatic Control* **53**(Special Issue), 122–125.

Mathur, H. & Datta, S. (2018), Design of state feedback control for a class of small scale structured descriptor systems, *in* 'IEEE Conference on Control Technology and Applications', Copenhagen, Denmark, pp. 1307–1312.

Moothedath, S., Chaporkar, P. & Belur, M. (2018), 'Minimum cost feedback selection for arbitrary pole assignment in structured systems', *IEEE Transactions on Automatic Control* **63**(11), 3881–3888.

Murota, K. (1987*a*), 'Refined study on structural controllability of descriptor systems by means of matroids', *SIAM Journal on Control and Optimization* **25**(4), 967–989.

Murota, K. (1987*b*), *Systems Analysis by Graphs and Matroids*, Springer-Verlag, Berlin, Germany.

Pasqualetti, F., Dorfler, F. & Bullo, F. (2013), 'Attack detection and identification in cyber-physical systems', *IEEE Transactions on Automatic Control* **58**(11), 2715–2729.

Ramos, G., Aguiar, A. P. & Pequito, S. (2022), 'An overview of structural systems theory', *Automatica* **140**, 1–26.

Reinschke, K. J. (1988), *Multivariable control: a graph-theoretic approach*, Springer-Verlag, Berlin, Germany.

Reinschke, K. J. (1994), 'Graph-theoretic approach to symbolic analysis of linear descriptor systems', *Linear Algebra and its Applications* **197**, 217–244.

Reinschke, K. J. & Wiedemann, G. (1997), 'Digraph characterization of structural controllability for linear descriptor systems', *Linear Algebra and its Applications* **266**, 199–217.

SageMath (n.d.), 'Sagemath mathematical software system'. **URL:** *https://www.sagemath.org/*

Sauer, P. W. & Pai, M. A. (1998), *Power System Dynamics and Stability*, Prentice Hall, Upper Saddle River, NJ.

Strang, G. (2006), *Linear Algebra and Its Applications*, fourth edn, Thomson, Belmont, CA.

Verghese, G. C., Lévy, B. C. & Kailath, T. (1981), 'A generalized state-space for singular systems', *IEEE Transactions on Automatic Control* **26**(4), 811–831.

Woude, J. V. D. & Murota, K. (1995), 'Disturbance decoupling with pole placement for structured systems: a graph theoretic approach', *SIAM Journal on Matrix Analysis and Applications* **16**(3), 922–942.

Yamada, T. & Luenberger, D. G. (1985), 'Generic controllability theorems for descriptor systems', *IEEE Transactions on Automatic Control* **30**(2), 144–152.

Zhou, T. (2022), 'Constructibility of a causal/impulse free NDS using descriptor form subsystems', *SCIENCE CHINA Information Sciences* **65**, 1–15.

Subashish Datta is an Assistant Professor in the Department of Electrical Engineering, Indian Institute of Technology (IIT) Delhi, India. He was an Assistant Professor in the School of Computing and Electrical Engineering, IIT Mandi, India. His research interests include control theory, robust control and LMIs, descriptor systems and multi-agent systems.

Harsha Mathur received the M.S. (By Research) degree in Control Systems from IIT Mandi, India in 2018. She is currently an engineer in a R& D industry, Bengaluru, India. Her research interests include descriptor systems and graph theory.