

Pole assignment problem: a structural investigation

AYLA ŞEFİK†† and M. EROL SEZER†

Based on the structure of a closed-loop system under a specified feedback pattern, a qualitative analysis of the problem of pole assignability is considered. The problem is first formulated algebraically, in terms of the relation $p = g(f)$ between the vector p of the closed-loop characteristic polynomial coefficients and the vector f of the non-zero elements of the feedback matrix. Then, translation to the structural framework is achieved by means of two theorems which give graph-theoretical sufficient conditions for solvability. These structural conditions also guarantee genericity of pole assignability.

1. Introduction

In the analysis of dynamical systems, such features as high dimensionality, uncertainty in system parameters, and constraints on information structure often lead to complications which cannot be solved by traditional methods. On the other hand, it may be possible to establish a way out through such problems after gaining sufficient insight into the structure of the system. This need for dealing with system structures is met by a qualitative analysis based on the structure of the system (structural analysis). This type of analysis is concerned with general properties of systems such as controllability, observability, existence of fixed modes, etc., which may also be regarded as the potential system properties. This is consistent with physical reality since system parameter values are never known precisely. The fact that digital computers work with true zeros and fuzzy numbers is another justification for this approach. Investigation of these properties from the genericity point of view is also of interest. A system is said to possess a property generically if that property holds for almost all values of the non-zero system parameters. In other words, if a property of a system is generic, then it fails to hold only in pathological cases when there is an exact matching of system parameters.

It was Lin (1974) who first introduced the concept in his characterization of structural controllability for single-input systems. His result was extended to the multi-input case by Shields and Pearson (1976). Sezer and Šiljak (1981) developed their characterization for structurally fixed modes in the same context.

This paper is concerned with a structural analysis of the problem of pole assignability. Non-zero system parameters are assumed to be algebraically independent and structural modelling based on structured matrices and directed graphs (digraphs) is used for system description. Graph-theoretic formulations due to Reinschke (1984) serve as tools in constructing our main results, which give graphical sufficient conditions for generic pole assignability by constant output feedback.

Section 2 is devoted to the formulation of the problem both algebraically and generically together with the establishment of the framework necessary for our

Received 15 May 1989. Revised 24 February 1991.

† Bilkent University, 06533 Bilkent, Ankara, Turkey.

‡ At present with Eastern Mediterranean University, North Cyprus.

structural approach. In § 3, we state and prove the two main theorems of the paper. Section 4 includes examples of classes of generically pole assignable systems that satisfy these results, thus demonstrating their non-triviality. A search algorithm to check the existence of a constant output feedback matrix which satisfies the conditions of the theorems is given in the Appendix.

2. Problem formulation and preliminaries

2.1. Algebraic formulation of the pole assignment problem

Consider a linear, time-invariant system described as

$$\mathcal{S}: \left. \begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \right\} \quad (1)$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^r$ are the state, input and output of \mathcal{S} , and A , B and C are real constant matrices of appropriate dimensions.

Let F be an $m \times r$ matrix with $v \leq mr$ non-zero elements which can be chosen arbitrarily. Applying a constant output feedback

$$\mathcal{F}: u = Fy \quad (2)$$

to \mathcal{S} in (1) results in a closed-loop system

$$\mathcal{S}(\mathcal{F}): \dot{x} = (A + BFC)x \quad (3)$$

having a characteristic polynomial

$$\begin{aligned} p(s) &= \det(sI - A - BFC) \\ &= s^n + p_1 s^{n-1} + \cdots + p_{n-1} s + p_n \end{aligned} \quad (4)$$

Let the non-zero, arbitrary elements of F be represented as a point $f = (f_1, f_2, \dots, f_v)$ in \mathbb{R}^v , and the coefficients of the characteristic polynomial $p(s)$ in (4) as a point $p = (p_1, p_2, \dots, p_n)$ in \mathbb{R}^n . Then, the relation between p and f can be represented by a smooth mapping $g: \mathbb{R}^v \rightarrow \mathbb{N}$ as

$$p = g(f) \quad (5)$$

where \mathbb{N} is a smooth manifold in \mathbb{R}^n . The pole-assignment problem is concerned with the existence of a solution $f \in \mathbb{R}^v$ of (5) for every given $p \in \mathbb{R}^n$.

We observe that $v \geq n$ is a necessary condition for solvability of (5) for all p , which we assume to hold in the rest of the paper. Let us, then, partition the feedback variables f_1, f_2, \dots, f_v into two disjoint subsets of f_v and f_c , respectively. Fixing the variables in f_c at particular real values, (5) is reduced to

$$p = \bar{g}(f_v) \quad (6)$$

where $\bar{g}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a restriction of g to \mathbb{R}^n . The following result (Reinschke 1987) gives a sufficient condition for pole assignability.

Lemma 1

Assume $n \leq v \leq mr$. If there exists a partitioning of the feedback variables f_1, f_2, \dots, f_v into two disjoint sets f_v and f_c containing n and $v - n$ elements such that

after appropriately fixing those in f_c the derivative \bar{g}_{f_v} is unimodular, then the system \mathcal{S} is arbitrarily pole assignable by the feedback \mathcal{F} . \square

Note that when \bar{g}_{f_v} is unimodular, then $\det \bar{g}_{f_v}$ is a constant so that \bar{g} is a homeomorphism; that is, for every $p \in \mathbb{R}^n$, there exists a unique $f_v \in \mathbb{R}^n$ satisfying $\bar{g}(f_v) = g(f_v, f_c) = p$.

Our main concern in this paper lies in a qualitative analysis of the pole assignment problem, based on the structure of the pair $(\mathcal{S}, \mathcal{F})$. In particular, we aim at deriving a structural counterpart of Lemma 1, and providing graph-theoretic conditions for generic pole assignability. We devote the rest of this section to establishing the framework needed for this approach.

2.2. Structural representation of systems

In this section, we review some basic concepts and results related to structured matrices, genericity (Shields and Pearson 1976), graph theory (Harary *et al.* 1965), and structural representation of systems (Sage 1977).

Structured matrices and generic properties

Two matrices $M_1, M_2 \in \mathbb{R}^{p \times q}$ are said to be 'structurally equivalent' if there is one-to-one correspondence between the locations of their non-zero entries. The equivalence class of structurally equivalent matrices in $\mathbb{R}^{p \times q}$ can be represented by a $p \times q$ 'structured matrix' \mathbf{M} , whose entries are either fixed zeros or algebraically independent parameters in \mathbb{R} . If \mathbf{M} has μ non-zero parameters, then associated with \mathbf{M} we define a parameter space \mathbb{R}^μ such that for every $d \in \mathbb{R}^\mu$, $\mathbf{M}(d)$ defines a fixed matrix in the equivalence class that \mathbf{M} represents. A fixed matrix M is said to be admissible with respect to \mathbf{M} , denoted as $M \in \mathbf{M}$, if $M = \mathbf{M}(d)$ for some $d \in \mathbb{R}^\mu$.

Let Π be a property asserted about the structured matrix \mathbf{M} . Then it is a mapping $\Pi: \mathbb{R}^\mu \rightarrow \{0, 1\}$ defined as

$$\Pi(d) = \begin{cases} 1, & \text{if } \Pi \text{ holds for } \mathbf{M}(d) \\ 0, & \text{otherwise} \end{cases}$$

Let $\Phi(d)$ be a polynomial in $d = (d_1, \dots, d_\mu)$ with real coefficients. The set

$$\Gamma = \{d \in \mathbb{R}^\mu \mid \Phi(d) = 0\}$$

is a 'variety' in \mathbb{R}^μ . Γ is said to be proper if $\Gamma \neq \mathbb{R}^\mu$ and non-trivial if $\Gamma \neq \emptyset$. The property Π is said to be 'generic' if there exists a proper variety Γ such that $\ker \Pi \subset \Gamma$. A generic property holds almost everywhere in \mathbb{R}^μ .

The 'generic rank' of a structured matrix \mathbf{M} , denoted by $\bar{\rho}(\mathbf{M})$, is the maximal rank $\mathbf{M}(d)$ can achieve for $d \in \mathbb{R}^\mu$. The set $\{d \in \mathbb{R}^\mu \mid \text{rank } \mathbf{M}(d) < \bar{\rho}(\mathbf{M})\}$ can easily be shown to be a proper variety in \mathbb{R}^μ , so that almost all fixed matrices $\mathbf{M}(d)$ have rank $\bar{\rho}(\mathbf{M})$.

Diagraphs

A 'diagraph' is an ordered pair $\mathcal{D} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is a finite set of 'vertices' and \mathcal{E} a set of oriented 'edges'. An edge oriented from $v_j \in \mathcal{V}$ to $v_i \in \mathcal{V}$ is denoted by the ordered pair (v_j, v_i) , where v_j is called the 'tail' and v_i the 'head' of the edge. If $(v_j, v_i) \in \mathcal{E}$, then v_j is said to be 'adjacent' to v_i , and v_i adjacent from v_j . The

adjacency relation can be described by a square binary matrix, $\mathbf{R} = (r_{ij})$ such that $r_{ij} = 1$ if and only if $(v_j, v_i) \in \mathcal{E}$. A sequence of edges $\{(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)\}$ where all vertices are distinct is called a ‘path’ from v_1 to v_k , denoted by (v_1, v_k) . In this case, v_k is said to be ‘reachable’ from v_1 . If v_k coincides with v_1 , then the path is called a ‘cycle’. The path that remains after the removal of an edge of a cycle is called the ‘complementary path’ of that edge relative to the cycle. Any two cycles are said to be ‘disjoint’ if they have no common vertices. A collection of disjoint cycles is called a ‘cycle family’.

A digraph $\mathcal{D}_s = (\mathcal{V}_s, \mathcal{E}_s)$, with a vertex set $\mathcal{V}_s = \{v_0, v_1, \dots, v_t\}$ and an edge set $\mathcal{E}_s = \{(v_0, v_1), (v_1, v_2), \dots, (v_{t-1}, v_t)\}$, is called a ‘stem’. v_0 and v_t are the ‘origin’ and the ‘tip’ of the stem, respectively. A digraph $\mathcal{D}_b = (\mathcal{V}, \mathcal{E}_b)$, with \mathcal{V} as above and $\mathcal{E}_b = \mathcal{E}_s \cup \{(v_0, v_1)\}$ is called a ‘bud’. v_0 is the origin and (v_0, v_1) is called the ‘distinguished edge’ of \mathcal{D}_b . A cactus is a digraph $\mathcal{D}_c = \mathcal{D}_s \cup \mathcal{D}_{b1} \cup \mathcal{D}_{b2}, \dots, \cup \mathcal{D}_{bk}$, where \mathcal{D}_s is a stem with origin v_0 and tip v_t ; and \mathcal{D}_{bi} are buds with origins $v_i \neq v_t$, such that v_i is the only vertex common to $\mathcal{D}_s \cup \mathcal{D}_{b1} \cup \mathcal{D}_{b2} \cup \dots \cup \mathcal{D}_{b,i-1}$ and \mathcal{D}_{bi} , $i = 1, \dots, k$. Origin v_0 and tip v_t of \mathcal{D}_s are also the origin and the tip of \mathcal{D}_c , respectively. If \mathcal{D}_s above is replaced by a bud, then the digraph is called a ‘precactus’, denoted by \mathcal{D}_p . Clearly, by deleting an appropriate edge of a precactus, it can be reduced to a cactus.

In a cactus, every vertex is reachable from the origin through a unique path. If in a cactus $\mathcal{D}_c = (\mathcal{V}_c, \mathcal{E}_c)$, vertices that are adjacent from the origin v_0 are v_1, v_2, \dots, v_q , then the sets $\mathcal{V}_i = \{v \in \mathcal{V} \mid v \text{ is reachable from } v_i\}$ are disjoint and $\mathcal{V}_c = \{v_0\} \cup \mathcal{V}_1 \cup \mathcal{V}_2 \cup \dots \cup \mathcal{V}_q$. Each of the subgraphs of \mathcal{D}_c defined by one of the vertex sets $\{v_0\} \cup \mathcal{V}_i$ is called a ‘bunch’ of the cactus. The bunch that contains the tip of the cactus is called the ‘terminal bunch’, and the others (if any) ‘non-terminal bunches’. Thus a terminal bunch is a cactus itself and a non-terminal bunch is a precactus.

System structure matrix and system digraph

Associated with the system \mathcal{S} of (1), we define a square structured matrix as

$$\mathbf{S} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} \\ \mathbf{C} & \mathbf{O} & \mathbf{O} \end{bmatrix} \tag{7}$$

which is called the ‘system structure matrix’. Viewing the matrix \mathbf{S} as a binary matrix with zero and non-zero elements, we define the digraph $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ which assumes \mathbf{S} as its adjacency matrix to be the digraph of the system \mathcal{S} . For convenience, the vertex set of \mathcal{D} can be partitioned as $\mathcal{V} = \mathcal{U} \cup \mathcal{X} \cup \mathcal{Y}$, where \mathcal{U} , \mathcal{X} and \mathcal{Y} are the sets of input, state and output variables, respectively. The system digraph \mathcal{D} completely characterizes the structure of \mathcal{S} . We say that two dynamic systems are structurally equivalent if their digraphs are the same up to an enumeration of \mathcal{U} , \mathcal{X} and \mathcal{Y} .

The subgraph $D_{ux} = (\mathcal{X} \cup \mathcal{U}, \mathcal{E}_{ux})$ obtained by removing from \mathcal{D} the output vertices and the edges connected to them is called the input-truncated system digraph and corresponds to the system structure matrix

$$\mathbf{S}_{ux} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{O} & \mathbf{O} \end{bmatrix} \tag{8}$$

When a feedback of the form (2) is applied to \mathcal{S} of (1), the resulting closed loop-system of (3) has the system structure matrix

$$\mathbf{S}(\mathbf{F}) = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{F} \\ \mathbf{C} & \mathbf{O} & \mathbf{O} \end{bmatrix} \quad (9)$$

Accordingly, the system digraph becomes $\mathcal{D}(\mathcal{F}) = (\mathcal{V}, \mathcal{E} \cup \mathcal{E}_{\mathcal{F}})$, where $\mathcal{E}_{\mathcal{F}} = \{(y_j, u_i) \mid f_{ij} \neq 0\}$ is the set of feedback edges.

For convenience, the edges in \mathcal{E} are called the ‘ d -edges’ and those in $\mathcal{E}_{\mathcal{F}}$ the ‘ f -edges’. Accordingly, a cycle is called an f -cycle if it contains at least one f -edge and a d -cycle otherwise. Similarly, a cycle family is called an f -cycle family if it contains at least one f -edge, a simple f -cycle family if it contains one and only one f -edge, and a d -cycle family otherwise. Note that if a feedback variable f_{ij} is given a fixed non-zero value, then the corresponding f -edge (y_j, u_i) becomes a d -edge as f_{ij} is no more different from a non-zero parameter of A , B or C .

A system \mathcal{S} is said to be ‘structurally controllable’ if it is either controllable or structurally equivalent to a controllable system. \mathcal{S} is structurally controllable if and only if the output-truncated system digraph D_{ux} is spanned by a family of disjoint cacti, that is, there exist a family of cacti $\mathcal{D}_{ci} = (\mathcal{V}_{ci}, \mathcal{E}_{ci})$ with $\mathcal{V}_{ci} = \{u_{k_i}\} \cup \mathcal{X}_i$ and $\mathcal{E}_{ci} \subset \mathcal{E}_{ux}$ such that $\cup \mathcal{X}_i = \mathcal{X}$.

2.3. Generic pole assignment problem

Imitating the definitions of the well-known structural properties such as structural controllability and existence of structurally fixed modes, we state the following definition.

Definition 1

A system \mathcal{S} of (1) is said to be structurally pole assignable by a feedback \mathcal{F} of (2) if there exists a system structurally equivalent to \mathcal{S} which is pole assignable by \mathcal{F} . □

Let us assume, as in an analysis of structural controllability that the non-zero parameters of the system structure matrix \mathbf{S} in (7) are algebraically independent, and correspond to the data point $d \in \mathbb{R}^{\mu}$. Then, the relation in (5) can be expressed as

$$p = g(d, f) \quad (10)$$

to indicate the dependence of g on the system parameters. Clearly, structural pole assignability is concerned with the existence of a particular data point $d^* \in \mathbb{R}^{\mu}$ for which the equation

$$p = g(d^*, f) = g^*(f) \quad (11)$$

has a solution for every given $p \in \mathbb{R}^n$. We note that solvability of (11) does not readily imply solvability of (10) for almost all $d \in \mathbb{R}^{\mu}$. In other words, structural pole assignability is not a generic property, or at least, cannot easily be proved to be a generic property. The reason is that the solvability of the non-linear equation (10) cannot easily be reduced to a condition involving only the parameter vector d .

We do, however, aim at obtaining structural conditions in terms of the system digraph, which guarantee genericity of structural pole assignability. For this purpose we refer to the formulation of Reinschke (1984), which is summarized below.

Consider the closed loop system digraph $\mathcal{D}(\mathcal{F}) = (\mathcal{V}, \mathcal{E} \cup \mathcal{E}_F)$ associated with the system structure matrix $\mathbf{S}(\mathbf{F})$ of (9). By assigning a weight to every edge, $\mathcal{D}(\mathcal{F})$ becomes a weighted digraph. The weight of a d -edge is the corresponding non-zero parameter value of A , B or C , and the weight of an f -edge is the corresponding variable feedback gain. Accordingly, the weight of a path, a cycle or a cycle family is the product of weights of all edges involved. Denoting the number of cycles in a cycle family \mathcal{CF} by $\sigma(\mathcal{CF})$, the weight of \mathcal{CF} by $\omega(\mathcal{CF})$, and defining the width $\gamma(\mathcal{CF})$ of \mathcal{CF} to be the total number of state vertices covered by \mathcal{CF} , Reinschke proved the following.

Lemma 2

The coefficients $p_k = g_k(f)$, $k = 1, 2, \dots, n$, of the closed-loop characteristic polynomial are given as

$$g_k(f) = \sum_{\gamma(\mathcal{CF})=k} (-1)^{\sigma(\mathcal{CF})} \omega(\mathcal{CF}) \quad (12)$$

where the summation is carried over all cycle families of width k . □

An immediate application of this lemma is that a feedback variable appears in a coefficient p_k of the closed-loop characteristic polynomial only if it takes part in a cycle family of width k .

3. Graphical conditions for generic pole assignability

We start by considering a special case of Lemma 1.

Corollary 1

Let f_v and f_c be as defined in Lemma 1, with the feedback variables in f_v renumbered as f_1, f_2, \dots, f_n . For a partitioning $\mathbb{N} = \mathbb{I} \cup (\mathbb{N} - \mathbb{I})$ with $\mathbb{I} \neq \emptyset$, of the index set $\mathbb{N} = \{1, 2, \dots, n\}$, define auxiliary variables \tilde{f}_k as

$$\tilde{f}_k = \begin{cases} f_k, & k \in \mathbb{I} \\ \theta_k f_k + \psi_k, & k \in \mathbb{N} - \mathbb{I} \end{cases} \quad (13)$$

where $\theta_k = \theta_k(d)$ is a non-zero polynomial in d , and $\psi_k = \psi_k(d, f_s)$ is a polynomial in f_l , $l \in \mathbb{I}$, with coefficients being polynomials in d . Suppose that the restriction \bar{g} of g in (6) to \mathbb{R}^n is given by

$$\bar{g}_k(d, f_v) = \bar{g}_k(d; \tilde{f}) = \alpha_k + \sum_{l=1}^n e_{kl} \tilde{f}_l, \quad k = 1, 2, \dots, n \quad (14)$$

where $\alpha_k = \alpha_k(d)$ and $e_{kl} = e_{kl}(d)$. Then, \mathcal{S} is generically pole assignable by \mathcal{F} if the coefficient matrix $E = E(d) = (e_{kl})$ has full generic rank. □

Proof

The derivative of \bar{g} is computed as

$$\bar{g}_{f_v} = E(d)\Xi(d, f_1)$$

where $\Xi = (\xi_{kl})$ has elements

$$\xi_{kl} = \begin{cases} 1, & k \in \mathbb{1}, l = k \\ 0, & k \in \mathbb{1}, l \neq k \\ \theta_k, & k \in \mathbb{N} - \mathbb{1}, l = k \\ 0, & k, l \in \mathbb{N} - \mathbb{1}, l \neq k \\ \partial\psi_k / \partial f_l, & k \in \mathbb{N} - \mathbb{1}, l \in \mathbb{1} \end{cases}$$

It follows that Ξ can be permuted into

$$\begin{bmatrix} I_{\mathbb{1}} & 0 \\ \partial\Psi / \partial f_{\mathbb{1}} & \Theta_{\mathbb{N} - \mathbb{1}} \end{bmatrix}$$

where $\Theta_{\mathbb{N} - \mathbb{1}} = \text{diag} \{ \theta_k, k \in \mathbb{N} - \mathbb{1} \}$, and $\partial\Psi / \partial f_{\mathbb{1}} = (\partial\psi_k / \partial f_l), k \in \mathbb{N} - \mathbb{1}, l \in \mathbb{1}$. Thus $\Xi(d, f)$ is generically unimodular, and the proof follows from Lemma 1. \square

We note that under the conditions of Corollary 1, the mapping \bar{g} can be decomposed as $\bar{g} = \tilde{g} \circ h$, where $\tilde{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the affine mapping defined in (14), and $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined in (13), both mappings being homeomorphisms. The significance of Corollary 1 lies in the fact that its assumptions and the full generic rank condition on the matrix E can be characterized, with the help of Lemma 2, in terms of the weighted closed-loop digraph $\mathcal{D}(\mathcal{F})$. This leads us to two main results which we state and prove below.

Theorem 1

Suppose that in $\mathcal{D}(\mathcal{F})$ there exists a choice of n distinct f -edges, renumbered conveniently as f_1, f_2, \dots, f_n , which after converting the remaining f -edges into d -edges by fixing their weights at arbitrary values, satisfy the following conditions:

- (i) no two f -edges occur in the same cycle;
- (ii) all f -cycles have a vertex in common;
- (iii) for $k = 1, 2, \dots, n$, there exist particular simple f -cycle families of width k , denoted by \mathcal{CF}_k^* , such that
 - (a) $f_k \in \mathcal{CF}_k^*$; and
 - (b) any other simple f -cycle family of width k which contains an f -edge $f_l, l \leq k$, also contains a d -edge which appears in no $\mathcal{CF}_j^*, j \leq k$.

Then \mathcal{S} is generically pole assignable with \mathcal{F} . \square

Proof

Conditions (i) and (ii) guarantee that every f -cycle family is a simple f -cycle family so that each product term $\omega(\mathcal{CF})$ in (12) contains at most one variable weight. In other words, each g_k in (12) is an affine function of f_1, f_2, \dots, f_n as in (14), so that \bar{g} has the structure in Corollary 1 with $\tilde{f}_k = f_k, k \in \mathbb{N}$, that is with $\mathbb{1} = \mathbb{N}$. Therefore, it suffices to show that the coefficient matrix $E = (e_{kl})$ in Corollary 1 is generically non-singular. For this, we first note that condition (iii)(a) implies that

each e_{kk} , $k \in \mathbb{N}$, contains at least one non-zero product corresponding to \mathcal{CF}_k^* , which we denote by e_{kk}^* . We now define $d_n = d$, $E_n(d_n) = E(d)$, and partition E_n as

$$E_n(d_n) = \left[\begin{array}{c|c} E_{n-1}(d_n) & e_{jn}(d_n)'s \\ \hline e_{ni}(d_n)'s & e_{nn}^*(d_n) + \hat{e}_{nn}(d_n) \end{array} \right] \tag{15}$$

where \hat{e}_{nn} denotes what is left from e_{nn} after separating e_{nn}^* (if there remains any). For a fixed $l \leq n$, either f_l appears in no cycle family of width n , in which case $e_{nl} \equiv 0$ or if it does, then by condition (iii)(b), the corresponding product term contains the weight of a d -edge, which occurs in no e_{kk}^* , $k \leq n$. Let d_{n-1} denote the parameter vector after all parameters corresponding to such d -edges are set to zero. Then $E_n(d_{n-1})$ is of the form

$$E_n(d_{n-1}) = \left[\begin{array}{c|c} E_{n-1}(d_{n-1}) & e_{jn}(d_{n-1})'s \\ \hline 0 & e_{nn}^*(d_{n-1}) \end{array} \right]$$

where $e_{nn}^*(d_{n-1})$ consists of a single non-zero product term, and each diagonal element $e_{kk}(d_{n-1})$ of $E_{n-1}(d_{n-1})$ still contains the product term $e_{kk}^*(d_{n-1}) = e_{kk}^*(d_n)$, $k = 1, 2, \dots, n - 1$. Obviously, $E_n(d_n)$ is generically non-singular if $E_n(d_{n-1})$ is. On the other hand, $E_n(d_{n-1})$ is generically non-singular if and only if $E_{n-1}(d_{n-1})$ is. Now, replacing d_n and $E_n(d_n)$ by d_{n-1} and $E_{n-1}(d_{n-1})$ and repeating the argument above, we come to the conclusion that $E_n(d_n)$ is generically non-singular if $E_1(d_1) = e_{11}^*(d)$ is non-zero, which is guaranteed by condition (iii)(a). This completes the proof. \square

A more general result, which makes full use of Corollary 1 is given by the following.

Theorem 2

The result of theorem 1 remains valid if condition (ii) is replaced by the following.

- (ii)' To any two f -edges f_p and f_q that appear in disjoint cycles there corresponds a unique pair of edges f_r and d_r such that
 - (a) d_r appears in every cycle of f_r but in no cycle of f_p or f_q , and
 - (b) to any two disjoint cycles \mathcal{C}_p and \mathcal{C}_q of f_p and f_q there corresponds a cycle \mathcal{C}_r of f_r which covers exactly the same state vertices as \mathcal{C}_p and \mathcal{C}_q cover, and vice versa. \square

Proof

The proof is based on the following facts.

Fact 1

$\mathcal{D}(\mathcal{F})$ does not contain more than two pairwise disjoint f -cycles. \square

Proof of Fact 1

Suppose that $\mathcal{D}(\mathcal{F})$ contains three pairwise disjoint f -cycles formed by the f -edges f_p, f_q and f_s . Let us denote, for convenience, the pair of edges f_r and d_r associated with each pair (f_i, f_j) , $i, j = p, q, s, i \neq j$, by f_{ij} and d_{ij} . Then, condition (ii)' implies that $\mathcal{D}(\mathcal{F})$ contains a subgraph which is isomorphic to one of the basic structures shown in Fig. 1. (There are eight possible combinations of different orientations of the edges f_{ij} , $i, j = p, q, s, i \neq j$, but six of these are essentially the same as one of the other two except for a relabelling of p, q and s .) However, each of these subgraphs contradicts condition (i), the one in Fig. 1(a) containing a cycle which includes three f -edges, f_{pq}, f_{sp} and f_{qs} , and the one in Fig. 1(b) containing a cycle which includes two f -edges f_{pq} and f_{qs} . Therefore, $\mathcal{D}(\mathcal{F})$ cannot contain three disjoint f -cycles. It cannot contain four or more pairwise disjoint f -cycles either, because this necessarily includes the existence of three pairwise disjoint f -cycles. This completes the proof of Fact 1. \square

Fact 2

The correspondence between the (f_r, d_r) 's and the pairs of (f_p, f_q) 's in the statement of condition (ii)' is one-to-one. \square

Proof of Fact 2

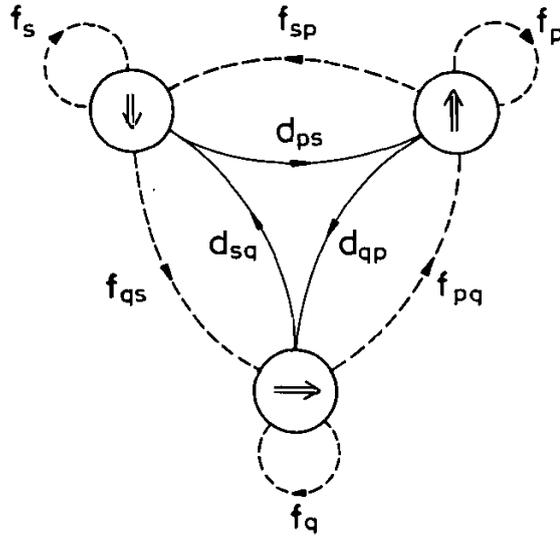
If (f_r, d_r) corresponds to two distinct pairs (f_p, f_q) and $(f_{p'}, f_{q'})$ then either all cycles formed by f_p and $f_{p'}$ or all cycles formed by f_q and $f_{q'}$ should cover the same state vertices. Suppose, without loss of generality, that the former is true and that $p < p'$. Since $f_{p'}$ appears in $\mathcal{CF}_{p'}^*$, which is of width p' , then so does f_p in some \mathcal{CF}_p^* of width p . However, every d -edge in \mathcal{CF}_p^* appears either in \mathcal{CF}_p^* or in $\mathcal{CF}_{p'}^*$, which violates condition (iii)(b). The situation is illustrated in Fig. 2, where $p = 1, p' = 2, \mathcal{CF}_p^* = \{d_1, d_2, f_p\}, \mathcal{CF}_{p'}^* = \{d_3, d_4, d_5, f_{p'}\}$ and $\mathcal{CF}_p = \{d_1, d_4, d_5, f_p\}$. \square

Fact 3

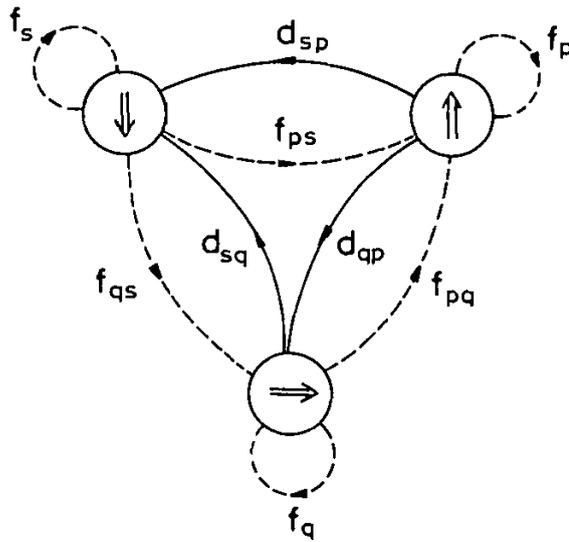
Suppose the pair (f_r, d_r) corresponds to the (unique) pair (f_p, f_q) . If f_r appears in a product term in some $g_k(f)$ of (12), then so does the product $f_p f_q$, and vice versa. Moreover, all the product terms that contain f_r in any $g_k(f)$ are of the form $e_{kr}(e_r f_r + e_{pq} f_p f_q)$, where e_{kr}, e_r , and e_{pq} are polynomials in d with e_r and e_{pq} being the same in all such expressions. \square

Proof of Fact 3

Let $\mathcal{C}_{r1}, \mathcal{C}_{r2}, \dots$, denote all simple f -cycles formed by f_r ; and for each i , let $\mathcal{CF}_{di1}, \mathcal{CF}_{di2}, \dots$, denote all d -cycle families which have no vertex in common with \mathcal{C}_{ri} . Then, any simple f -cycle family containing f_r is of the form $\mathcal{CF}_r = \mathcal{C}_{ri} \cup \mathcal{C}_{dij}$ for some i and j , so that $\omega(\mathcal{CF}_r) = \omega(\mathcal{C}_{ri}) \times \omega(\mathcal{C}_{dij}) = e_{ri} f_r e_{dij}$. By condition (ii)', to every \mathcal{C}_{ri} there correspond disjoint simple f -cycles \mathcal{C}_{pi} and \mathcal{C}_{qi} formed by f_p and f_q , which are also disjoint from all \mathcal{C}_{dij} . Therefore, they form an f -cycle family $\mathcal{CF}_{pq} = \mathcal{C}_{pi} \cup \mathcal{C}_{qi} \cup \mathcal{C}_{dij}$ of the same width as that of \mathcal{CF}_r , and having the weight $\omega(\mathcal{CF}_{pq}) = e_{pi} f_p \times e_{qi} f_q \times e_{dij}$. This shows the existence of the product $f_p f_q$ wherever f_r appears. The converse is also true, and the proof of the first part is



(a)



(b)

Figure 1. The two basic structures mentioned in the proof of Fact 1.

complete. Now, let e_r be the product of the weights of the d -edges which are common to all C_{ri} , but do not occur in some $\mathcal{C}_{pi} \cup \mathcal{C}_{qi}$. Obviously, d_r appears in e_r , so that $e_{ri} = e'_{ri} \times e_p$. Also define e_p and e_q to be the products of the weights of the d -edges which are common to all \mathcal{C}_{pi} and \mathcal{C}_{qi} , respectively, and which do not appear in some C_{ri} , and therefore write $e_{pi} = e'_{pi} \times e_p$ and $e_{qi} = e'_{qi} \times e_q$. Since for fixed i , $\mathcal{C}_{pi} \cup \mathcal{C}_{qi}$ and \mathcal{C}_{ri} cover exactly the same state vertices, then e_p and e_q may

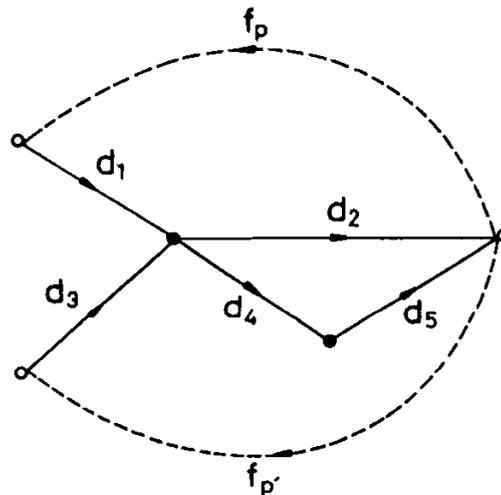


Figure 2. Illustration of the situation mentioned in the proof of Fact 2.

only contain weights of d -edges that are adjacent either from the input or to the output associated with f_p and f_q , respectively. Furthermore, $e'_{ri} = e'_{pi} \times e'_{qi}$. Then, $\omega(\mathcal{CF}_r) + \omega(\mathcal{CF}_{pq}) = e'_{ri}e_{dij}(e_r f_r + e_p e_q f_p f_q)$ independent of the widths of the cycle families \mathcal{CF}_r and \mathcal{CF}_{pq} , and the proof follows. \square

Now, returning to the proof of Theorem 2, Fact 1 together with condition (i) imply that each product term $\omega(\mathcal{CF})$ in (12) contains at most two variable weights. Also, defining

$$\mathbb{I} = \{k \mid f_k \text{ forms a cycle which is disjoint from some other } f\text{-cycle}\}$$

and \vec{f}_r as in (13) with $\theta_r = e_r$ and $\psi_r = e_{pq} f_p f_q$, Fact 3 guarantees the structure in (12). The rest of the proof is the same as that of Theorem 1. \square

The usefulness of Theorems 1 and 2 depends largely on the choice of the n feedback gains to be included in f_v , as well as on the choice of the zero or non-zero fixed values to be assigned to the remaining feedback gains in f_c . An algorithm, which determines whether such a choice of n feedback edges that satisfy the conditions of Theorem 2 exists, is given in the Appendix.

4. Examples of generically pole assignable systems

In this section we show that certain classes of systems which are known to be generically pole assignable by state or dynamic output feedback satisfy the conditions of Theorem 2 and thus demonstrate that Theorem 2 characterizes a non-trivial class of pole assignable structures.

4.1. Structurally controllable systems with state feedback

Consider a system described by

$$\mathcal{S}: \dot{x} = Ax + Bu \tag{16}$$

and the full state feedback law

$$\mathcal{F}: u = Fx \quad (17)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Since \mathcal{F} is a special case of static output feedback with states considered as outputs, the resulting closed-loop system $\mathcal{S}(\mathcal{F})$ can be represented by the reduced system structure matrix

$$S(F) = \begin{bmatrix} A & B \\ F & 0 \end{bmatrix} \quad (18)$$

Let the corresponding open- and closed-loop system digraphs be $\mathcal{D}_{ux} = (\mathcal{X} \cup \mathcal{U}, \mathcal{E}_{ux})$ and $\mathcal{D}_{ux}(\mathcal{F}) = (\mathcal{X} \cup \mathcal{U}, \mathcal{E}_{ux} \cup \mathcal{E}_F)$. We now state our main result concerning $\mathcal{S}(\mathcal{F})$.

Theorem 3

The following are equivalent:

- (a) \mathcal{S} is structurally controllable;
- (b) $\mathcal{S}(\mathcal{F})$ is generically pole-assignable;
- (c) there exists a choice of n feedback edges such that when the remaining feedback edges are assigned suitable fixed weights, $\mathcal{D}_{ux}(\mathcal{F})$ satisfies the conditions of Theorem 1. \square

The proof of Theorem 3 is based on the following two lemmas.

Lemma 3

Let $\mathcal{D}_c = (\mathcal{X} \cup \{u\}, \mathcal{E})$ be a cactus. Then there exists an enumeration of the state vertices such that

- (a) if x_i is on a non-terminal bunch and x_j is on the terminal bunch, then $i < j$;
- (b) if $(x_i, x_j) \in \mathcal{E}$ and x_j is not the tail of the distinguished edge of some bud, then $j = i + k + 1$, where k is the total number of state vertices on the precactus with origin x_i . \square

Proof

Using a modified depth-first search algorithm (Tarjan 1972), scan first the non-terminal bunches (if there are any) in any order, and last the terminal bunch of \mathcal{D}_c , and assign the integers $1, 2, \dots, n$ to the state vertices during the scanning process according to the following simple recursive scheme. Let the current vertex being visited be x_i . If there is a cactus or precactus with origin at x_i , then replace \mathcal{D}_c by this cactus or precactus (with x_i taking the role of u) and repeat. Otherwise, let the unique vertex adjacent from x_i be x^* . If x^* is not yet assigned an integer, let $i \leftarrow i + 1$, $x_i = x^*$, and repeat. Otherwise, x^* should be adjacent from the root of the cactus currently being scanned. Continue with another bunch. \square

It is obvious that this scanning of \mathcal{D}_c results in an enumeration of the state vertices which satisfies the requirements. To illustrate the scheme, enumeration of the vertices of a simple cactus is shown in Fig. 3.

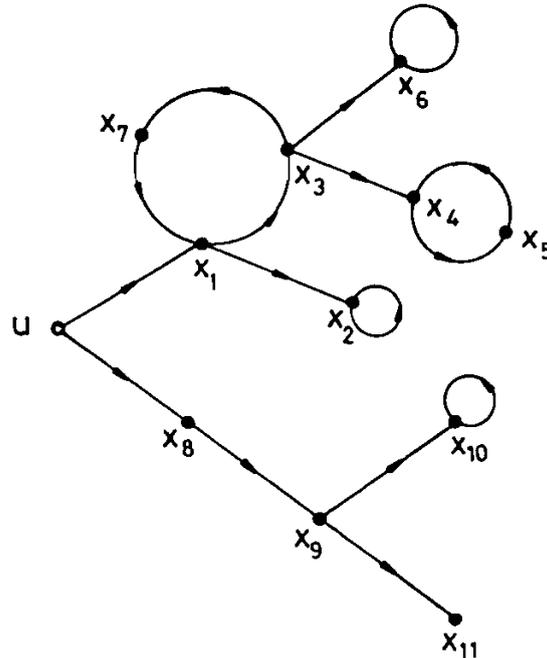


Figure 3. Enumeration of the state vertices in a cactus.

Lemma 4

Let $\mathbf{S} = (\mathbf{A}, \mathbf{B})$ be structurally controllable. Then there exists a fixed feedback matrix F_1 and a column \mathbf{b}_i of \mathbf{B} such that

- (a) the non-zero elements of $\mathbf{A} + \mathbf{B}F_1$ and \mathbf{b}_i are algebraically independent, and
- (b) the system $\mathbf{S}_1 = (\mathbf{A} + \mathbf{B}F_1, \mathbf{b}_i)$ is structurally controllable. □

Proof

If $(\mathbf{A}, \mathbf{b}_i)$ is structurally controllable for some i , let $F_1 = 0$. Otherwise, let $\mathcal{D}_{u,x}$ be spanned by a union of cacti $\mathcal{D}_{c1}, \mathcal{D}_{c2}, \dots, \mathcal{D}_{ck}$ with roots $u_{i_1}, u_{i_2}, \dots, u_{i_k}$ and tips $x_{n_1}, x_{n_1+n_2}, \dots, x_{n_1+\dots+n_k}$, where $1 \leq k \leq m$, $1 \leq i_1 < \dots < i_k \leq m$, and $n_1 + n_2 + \dots + n_k = n$. Let $F_1 = (f_{pq})$ with

$$f_{pq} = \begin{cases} 1, & \text{if } p = i_l, q = n_1 + \dots + n_{l-1}, \text{ for some } 2 \leq l \leq k \\ 0, & \text{otherwise} \end{cases}$$

and let $i = i_1$. Then, since the elements of (\mathbf{A}, \mathbf{B}) are algebraically independent and non-zero elements of F_1 are fixed as unity, the elements of $(\mathbf{A} + \mathbf{B}F_1, \mathbf{b}_i)$ are also algebraically independent. Moreover, \mathbf{S}_1 is spanned by a cactus obtained by making the roots of $\mathcal{D}_{c,l+1}$ coincide with $x_{n_1+\dots+n_l}$, $l = 1, \dots, k - 1$. □

Note that Lemma 4 is a structural counterpart of the well-known algebraic result (Davison and Wang 1973) that if (A, B) is controllable, then for almost all F_1 , $(A + BF_1, b_i)$ are controllable.

We now prove Theorem 3.

Proof of Theorem 3

Owing to Lemma 4, it suffices to give the proof for the single-input case.

(a) \Leftrightarrow (b): Obvious.

(c) \Rightarrow (b): Theorem 1.

(a) \Rightarrow (c): Let the system digraph \mathcal{D}_{ux} be spanned by a cactus \mathcal{D}_c , whose state vertices are enumerated as in Lemma 3. Let the feedback edges be enumerated in the same way so that $f_i = (x_i, u)$, $i = 1, 2, \dots, n$. Since all f -cycles in $\mathcal{D}_{ux}(\mathcal{F})$ pass through vertex u , conditions (i) and (ii) of Theorem 1 are readily satisfied. The enumeration of the state vertices guarantees that for $i = 1, 2, \dots, n$, any state vertex x_j with $j \leq i$ either lies on the complementary path of f_i in $\mathcal{D}_c(\mathcal{F})$, and hence belongs to the f -cycle defined by f_i , or belongs to a d -cycle in $\mathcal{D}_c(\mathcal{F})$ which has no vertex in common with the complementary path of f_i . Let \mathcal{CF}_i^* denote the union of these cycles in $\mathcal{D}(\mathcal{F})$. Obviously, \mathcal{CF}_i^* is a simple f -cycle family of width i which contains f_i . (For example, referring to Fig. 3, \mathcal{CF}_6^* consists of the f -cycle $\{(u, x_1), (x_1, x_3), (x_3, x_6), (x_6, u)\}$ and the d -cycles $\{(x_2, x_2)\}$ and $\{(x_4, x_5), (x_5, x_4)\}$.) This proves condition (iii)(a) of Theorem 1. Now, let \mathcal{CF}_i be any simple f -cycle family of width i which includes an f -edge f_j for some $j < i$. If \mathcal{CF}_i contains a d -edge which does not belong to the edge set of \mathcal{D}_c , then this edge does not appear in any \mathcal{CF}_i^* , and condition (iii)(b) of Theorem 1 is readily satisfied for \mathcal{CF}_i . Suppose all the d -edges of \mathcal{CF}_i belong to \mathcal{D}_c . Since \mathcal{CF}_i covers exactly i vertices, it covers a vertex x_k with $k \geq i$. Then, the edge originating from x_k in \mathcal{CF}_i is a d -edge (the only f -edge in \mathcal{CF}_i is f_j which originates from x_j) which does not appear in any \mathcal{CF}_i^* , $l \leq k$. Again, (iii)(b) is satisfied. This completes the proof. \square

4.2. *A class of structurally controllable and observable systems with dynamic output feedback*

Consider a single-input/single-output system

$$\mathcal{S}: \begin{cases} \dot{x} = Ax + bu \\ y = c^T x \end{cases} \tag{19}$$

to be controlled by a dynamic output feedback of the form

$$\mathcal{P}: \begin{cases} \dot{\hat{x}} = \hat{A}\hat{x} + \hat{b}y \\ u = \hat{c}^T \hat{x} + \hat{f}y \end{cases} \tag{20}$$

where $\hat{x} \in \mathbb{R}^n$ is the state of the controller \mathcal{P} . It is well known (Davison and Chatterjee 1971) that the closed-loop system consisting of \mathcal{S} and \mathcal{P} is the same as the one obtained by applying a constant output feedback of the form

$$\mathcal{F}_a: \begin{bmatrix} u \\ u_c \end{bmatrix} = \begin{bmatrix} \hat{f} & \hat{c}^T \\ \hat{b} & \hat{A} \end{bmatrix} \begin{bmatrix} y \\ y_c \end{bmatrix} \tag{21}$$

to an augmented system

$$\mathcal{S}_a: \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} b & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ u_c \end{bmatrix}$$

$$\begin{bmatrix} y \\ y_c \end{bmatrix} = \begin{bmatrix} c^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$
(22)

Thus the pole assignment problem by dynamic output feedback is essentially the same as pole assignment by constant output feedback, and hence can be attacked with the graph-theoretic approach of § 3.

We assume that \mathcal{S} is structurally controllable and observable, that is, it has no structurally fixed modes. Let $\mathcal{D}(f)$ be the digraph of the closed-loop system consisting of \mathcal{S} and the (scalar) constant output feedback

$$\mathcal{F}: u = fy$$

Then, \mathcal{S} having no structurally fixed modes is equivalent to the following two conditions (Pichai *et al.* 1984):

- (a) $\mathcal{D}(f)$ contains a cycle family \mathcal{CF} of width n ;
- (b) $\mathcal{D}(f)$ is strongly connected, i.e. each state vertex reaches every other either in \mathcal{D} , or through the feedback edge (y, u) .

We further assume that each cycle in \mathcal{CF} has a vertex in common with some input-output path in \mathcal{D} . This is a crucial assumption that enables us to define the auxiliary variables \hat{f}_k in (13) using simple polynomials ψ_k as will become clear in the following development.

We now choose the order of the controller \mathcal{P} to be $\hat{n} = n - 1$, and fix its structure as

$$\left. \begin{aligned} \hat{A} &= \begin{bmatrix} 0 & 0 & \dots & 0 & \hat{a}_{n-1} \\ 1 & 0 & \dots & 0 & \hat{a}_{n-2} \\ 0 & 1 & \dots & 0 & \hat{a}_{n-3} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & \hat{a}_1 \end{bmatrix}, & \hat{b} &= \begin{bmatrix} \hat{b}_{n-1} \\ \hat{b}_{n-2} \\ \vdots \\ \hat{b}_1 \end{bmatrix} \\ \hat{c}^T &= [0 \quad 0 \quad \dots \quad 0 \quad 1], & \hat{f} &= f \end{aligned} \right\} \quad (23)$$

where $\hat{a}_i, \hat{b}_i, i = 1, 2, \dots, n - 1$, and \hat{f} are variable feedback gains. Thus, of the n^2 elements of \mathcal{F}_a in (21), $n^2 - (2n - 1)$ are fixed at 0 or 1 with the remaining $2n - 1$ left as variable parameters.

With \mathcal{P} chosen as above, the closed-loop digraph $\mathcal{D}_a(\mathcal{F}_a)$ which corresponds to the system $\mathcal{S}_a(\mathcal{F}_a)$ has the structure shown in Fig. 4.

We now prove the following result about the pole assignability of $\mathcal{S}_a(\mathcal{F}_a)$.

Theorem 4

Suppose that $\mathcal{D}(f)$ contains a cycle family of width n , each cycle of which has a vertex in common with some input-output path in \mathcal{D} . Then $\mathcal{D}_a(\mathcal{F}_a)$ satisfies the conditions of Theorem 2 with n replaced by $n_a = 2n - 1$. □

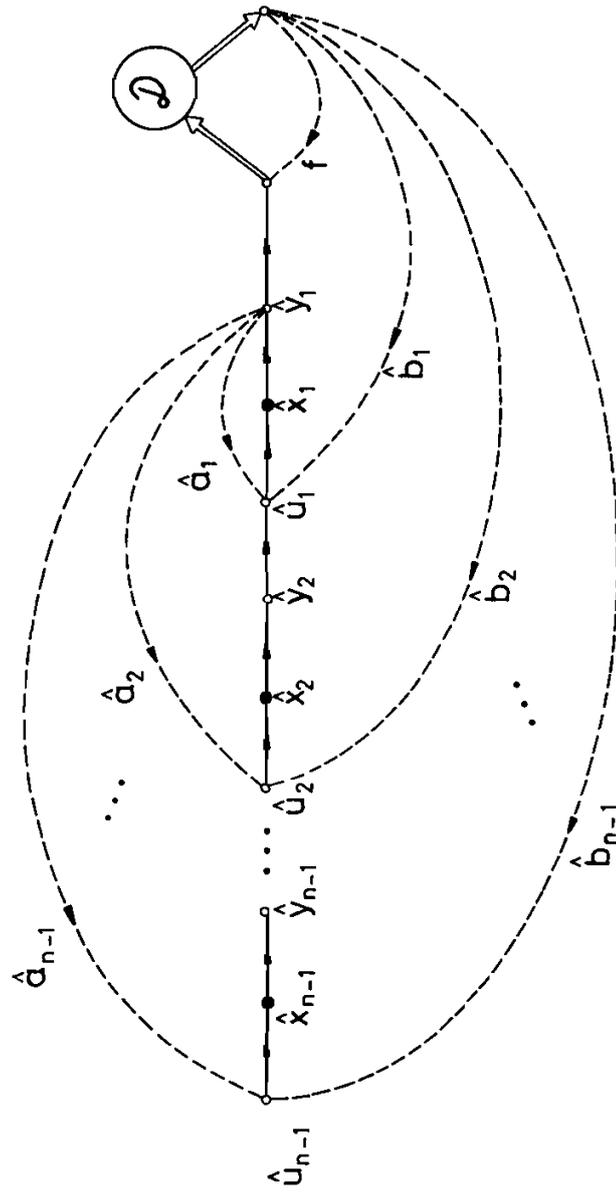


Figure 4. Illustration of the closed-loop system digraph $\mathcal{G}_a(\mathcal{F}_a)$.

Proof

Referring to Fig. 4, we first note that $\mathcal{D}_a(\mathcal{F}_a) = (\mathcal{V} \cup \hat{\mathcal{V}}, \mathcal{E} \cup \hat{\mathcal{E}} \cup \mathcal{E}_f)$, where $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is the digraph of \mathcal{S} , $\hat{\mathcal{D}} = (\hat{\mathcal{V}}, \hat{\mathcal{E}})$ is the digraph associated with the fixed parameters of $\hat{\mathcal{S}}$, and \mathcal{E}_f is the set of (variable) f -edges corresponding to the feedback parameters $\hat{a}_i, \hat{b}_i, i = 1, 2, \dots, n - 1$; and \hat{f} . Thus $\mathcal{D}_a(\mathcal{F}_a)$ has $n_a = 2n - 1$ state vertices, which is exactly the same as the number of f -edges. We show that these f -edges can be suitably ordered so as to satisfy the conditions of Theorem 2.

We first observe that f -cycles in $\mathcal{D}_a(\mathcal{F}_a)$ are of one of the following forms:

$$\mathcal{C}_I = \{(y, u), \overline{(u, y)}\}$$

$$\mathcal{C}_{II} = \{(\hat{y}_1, \hat{u}_i), \overline{(\hat{u}_i, \hat{y}_1)}\}$$

$$\mathcal{C}_{III} = \{(y, \hat{u}_i), \overline{(\hat{u}_i, \hat{y}_1)}, (\hat{y}_1, u), \overline{(u, y)}\}$$

where $\overline{(u, y)}$ denotes a path in \mathcal{D} and $\overline{(\hat{u}_i, \hat{y}_1)}$ denotes a path in $\hat{\mathcal{D}}$. Consequently, no f -cycle in $\mathcal{D}_a(\mathcal{F}_a)$ contains more than one f -edge, satisfying condition (i) of Theorem 2. Also, only $f_p = (y, u)$ and an f -edge $f_{qi} = (\hat{y}_1, \hat{u}_i), i = 1, 2, \dots, n - 1$, can appear in disjoint f -cycles (of forms \mathcal{C}_I and \mathcal{C}_{II} , respectively). It is not difficult to see that for every such pair $(f_p, f_{qi}), d_r = (\hat{y}_1, u)$ and $f_{ri} = (y, \hat{u}_i)$ form a unique pair which satisfies condition (ii)' of Theorem 2.

To continue the proof we need the following result.

Fact 4

$\mathcal{D}(f)$ has a subgraph $\tilde{\mathcal{D}}$ with the following properties:

- (a) $\tilde{\mathcal{D}}$ contains a unique cycle family $\tilde{\mathcal{C}}_{\mathcal{F}}$ of width n .
 - (b) Each cycle in $\tilde{\mathcal{C}}_{\mathcal{F}}$ has a vertex in common with some input-output path.
 - (c) $\tilde{\mathcal{D}}$ is minimal in the sense that removal of any edge violates (a) or (b) above.
-

Proof of Fact 4

Pick an arbitrary cycle family $\tilde{\mathcal{C}}_{\mathcal{F}}$ of width n in $\mathcal{D}(f)$, and a minimal set $\tilde{\mathcal{E}}_a$ of additional d -edges such that each cycle in $\tilde{\mathcal{C}}_{\mathcal{F}}$ has a vertex in common with some input-output path in the subgraph $\tilde{\mathcal{D}}$ formed by $\tilde{\mathcal{C}}_{\mathcal{F}}$ and these additional d -edges. Include f into $\tilde{\mathcal{D}}$, if not already included. If $\tilde{\mathcal{D}}$ contains another cycle family $\mathcal{C}_{\mathcal{F}}$ of width n , then one of the cycles in $\mathcal{C}_{\mathcal{F}}$ contains a d -edge which is not included in $\tilde{\mathcal{C}}_{\mathcal{F}}$. The subgraph $\tilde{\mathcal{D}}$ of $\tilde{\mathcal{D}}$ obtained by removing this particular d -edge still contains a cycle family of width n each cycle of which has a vertex in common with some input-output path. Replace $\tilde{\mathcal{D}}$ by $\tilde{\mathcal{D}}$, $\tilde{\mathcal{C}}_{\mathcal{F}}$ by $\mathcal{C}_{\mathcal{F}}$, and repeat the same argument. Each time by deleting a d -edge from $\tilde{\mathcal{D}}$ and modifying $\tilde{\mathcal{C}}_{\mathcal{F}}$, we eventually obtain a subgraph which satisfies properties (a) and (b). Finally, removing some d -edges from $\tilde{\mathcal{E}}_a$ if not needed for (b), minimality of $\tilde{\mathcal{D}}$ with respect to the properties (a) and (b) is guaranteed. This completes the proof. □

We note that $\tilde{\mathcal{D}}$ in Fact 4 may or may not contain the f -edge (y, u) . We continue with the proof of Theorem 3 by considering the two cases separately.

Case I: $\tilde{\mathcal{D}}$ does not include the f -edge (y, u)

In this case, \mathcal{CF} is a d -cycle family of width n . Let $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$ be the digraph obtained from $\mathcal{D}_a(\mathcal{F}_a)$ by replacing \mathcal{D} with $\tilde{\mathcal{D}}$. Since $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$ is obtained from $\mathcal{D}_a(\mathcal{F}_a)$ by removing some d -edges of \mathcal{D} , it suffices to complete the proof for $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$, because $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$ still satisfies conditons (i) and (ii)' of Theorem 2, and if it also satisfies condition (iii), then so does $\mathcal{D}_a(\mathcal{F}_a)$.

Let $\{\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_s\}$ be a family of input-output paths in $\tilde{\mathcal{D}}$ such that any d -cycle in \mathcal{CF} has a vertex in common with some $\tilde{\mathcal{P}}_j$. Define \mathcal{CF}_j to be the subfamily of all cycles in \mathcal{CF} which has no vertex in common with any $\tilde{\mathcal{P}}_l, l > j$. The definition of $\tilde{\mathcal{P}}_j$ and \mathcal{CF}_j is illustrated in Fig. 5 for a simple digraph $\tilde{\mathcal{D}}$. Note that $\mathcal{CF}_0 = \emptyset, \mathcal{CF}_s = \mathcal{CF}$, and $\mathcal{CF}_{j-1} \subset \mathcal{CF}_j, j = 1, 2, \dots, s$. We further define the integers α_j and β_j as the number of state vertices in $\tilde{\mathcal{P}}_j$ and \mathcal{CF}_j , i.e. $\alpha_j = \gamma(\tilde{\mathcal{P}}_j)$ and $\beta_j = \gamma(\mathcal{CF}_j), j = 1, 2, \dots, s$, and let $\alpha_{s+1} = \beta_{-1} = \beta_0 = 0$ for convenience. It is easy to see that α_j and β_j satisfy

$$(a) \quad 1 \leq \beta_1 < \beta_2 < \dots < \beta_s = n$$

$$(b) \quad \alpha_j + \beta_{j-1} \leq \alpha_{j+1} + \beta_j - 1, \quad 1 \leq j \leq s$$

We partition the integers $\{1, 2, \dots, n_a = 2n - 1\}$ into two groups at s levels as shown in Table 1, where Group A /Level 0 is empty if $\alpha_1 = 1$, and Group A /Level s is empty if $\alpha_s + \beta_{s-1} = n$.

| Level | Group A | Group B |
|----------|--|--|
| 0 | $1, \dots, \alpha_1 - 1$ | $\alpha_1, \dots, \alpha_1 + \beta_1 - 1$ |
| 1 | $\alpha_1 + \beta_1, \dots, \alpha_2 + 2\beta_1 - 1$ | $\alpha_2 + 2\beta_1, \dots, \alpha_2 + \beta_1 + \beta_2 - 1$ |
| \vdots | \vdots | \vdots |
| p | $\alpha_p + \beta_{p-1} + \beta_p, \dots, \alpha_{p+1} + 2\beta_p - 1$ | $\alpha_{p+1} + 2\beta_p, \dots, \alpha_{p+1} + \beta_p + \beta_{p+1} - 1$ |
| \vdots | \vdots | \vdots |
| $s - 1$ | $\alpha_{s-1} + \beta_{s-2} + \beta_{s-1}, \dots, \alpha_s + 2\beta_{s-1} - 1$ | $\alpha_s + 2\beta_{s-1}, \dots, \alpha_s + \beta_{s-1} + \beta_s - 1$ |
| s | $\alpha_s + \beta_{s-1} + \beta_s, \dots, 2n - 1$ | |

Table 1.

We now define the feedback edges f_k and the associated cycle families \mathcal{CF}_k^* , $1 \leq k \leq 2n - 1$, for $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$ as follows.

(a) If $k \in$ Group A /Level p , i.e. if

$$\alpha_p + \beta_{p-1} + \beta_p \leq k \leq \alpha_{p+1} + 2\beta_p - 1$$

then

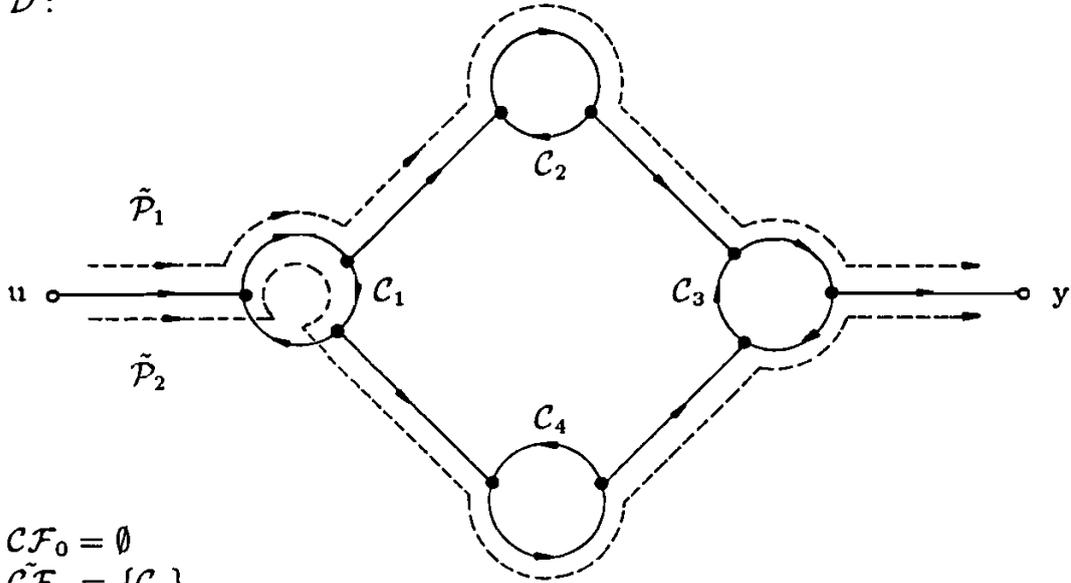
$$f_k = (\hat{y}_1, \hat{u}_{k-\beta_p})$$

$$\mathcal{CF}_k^* = \overline{\{(\hat{u}_{k-\beta_p}, \hat{y}_1), (\hat{y}_1, \hat{u}_{k-\beta_p})\}} \cup \mathcal{CF}_p$$

(b) If $k \in$ Group B /Level p , i.e. if

$$\alpha_{p+1} + 2\beta_p \leq k \leq \alpha_{p+1} + \beta_p + \beta_{p+1} - 1$$

$\tilde{\mathcal{D}}$:



$$\begin{aligned} \mathcal{CF}_0 &= \emptyset \\ \tilde{\mathcal{CF}}_1 &= \{C_2\} \\ \tilde{\mathcal{CF}}_2 &= \mathcal{CF} = \{C_1, C_2, C_3, C_4\} \end{aligned}$$

Figure 5. Definition of $\tilde{\mathcal{P}}_j$ and $\tilde{\mathcal{CF}}_j$.

then

$$\begin{aligned} f_k &= (y, \hat{u}_{k-\alpha_{p+1}-\beta_p}) \\ \tilde{\mathcal{CF}}_k^* &= \overline{\{(\hat{u}_{k-\alpha_{p+1}-\beta_p}, \hat{y}_1), (\hat{y}_1, u), \tilde{\mathcal{P}}_{p+1}, (y, \hat{u}_{k-\alpha_{p+1}-\beta_p})\}} \cup \tilde{\mathcal{CF}}_p \end{aligned}$$

Note that in case (a)

$$\begin{aligned} \gamma(\tilde{\mathcal{CF}}_k^*) &= \gamma(\overline{(\hat{u}_{k-\beta_p}, \hat{y}_1)}) + \gamma(\tilde{\mathcal{CF}}_p) \\ &= (k - \beta_p) + \beta_p = k \end{aligned}$$

and in case (b)

$$\begin{aligned} \gamma(\tilde{\mathcal{CF}}_k^*) &= \gamma(\overline{(\hat{u}_{k-\alpha_{p+1}-\beta_p}, \hat{y}_1)}) + \gamma(\tilde{\mathcal{P}}_{p+1}) + \gamma(\tilde{\mathcal{CF}}_p) \\ &= (k - \alpha_{p+1} - \beta_p) + \alpha_{p+1} + \beta_p = k \end{aligned}$$

so that $\tilde{\mathcal{CF}}_k^*$ is an f -cycle family of width k in $\tilde{\mathcal{D}}_a(\mathcal{F}_a)$. By definition, it includes f_k and no other f -edge, satisfying condition (iii)(a) of Theorem 2. Finally, to prove condition (iii)(b), let \mathcal{CF}_k be a simple f -cycle family of width k , which includes some f -edge f_l with $l \leq k$. We consider all possibilities for k and l , as follows.

(1) $k \in \text{Group } A \text{ or } B/\text{Level } p, l \in \text{Group } A/\text{Level } q, q \leq p$. In this case, $\tilde{\mathcal{CF}}_k = \{(\hat{u}_{l-\beta_q}, \hat{y}_1), (\hat{y}_1, \hat{u}_{l-\beta_k})\} \cup \mathcal{CF}_d$, where \mathcal{CF}_d is a d -cycle family in \mathcal{D} . If \mathcal{CF}_d contains a d -edge which does not belong to $\tilde{\mathcal{CF}}_p$ (remember that $k \in \text{Level } p$), then that d -edge does not belong to any $\mathcal{CF}_r, r \leq p$, either. Since any $j \leq k$ is at some level $r \leq p$, and $\tilde{\mathcal{CF}}_j^*$ includes \mathcal{CF}_r , this particular d -edge appears in no $\tilde{\mathcal{CF}}_k^*, j \leq k$, and condition (iii)(b) is satisfied. If \mathcal{CF}_d does not contain such a d -edge, then minimality of $\tilde{\mathcal{D}}$ implies $\mathcal{CF}_d \subset \tilde{\mathcal{CF}}_p$. Then, $\gamma(\mathcal{CF}_d) \leq \beta_p$, and we must have

| T | Level | Group A | Group B |
|---|-------|---|--|
| | s | $\alpha_s + \beta_{s-1} + \beta_s, \dots, \alpha_{s+1} + \beta_s + \beta_s - 1$ | $\alpha_{s+1} + \beta_s + \beta_s, \dots, 2n - 1,$ |

Table 2.

$\gamma(\tilde{\mathcal{CF}}_k) \leq l - \beta_q + \beta_p$, with equality holding only if $\tilde{\mathcal{CF}}_d = \tilde{\mathcal{CF}}_p$. This, however, is impossible because

(a) if $q = p$, then either $\gamma(\tilde{\mathcal{CF}}_k) < l \leq k$ or $\gamma(\tilde{\mathcal{CF}}_k) \leq l < k$ (because $\gamma(\tilde{\mathcal{CF}}_k) = l = k$ can occur only if $\tilde{\mathcal{CF}}_k = \tilde{\mathcal{CF}}_k^*$);

(b) if $q < p$, then $l - \beta_q + \beta_p \leq \alpha_{q+1} + \beta_q + \beta_p - 1 < \alpha_{p+1} + \beta_p + \beta_p - 1 \leq k - 1$,

contradicting the assumption that $\gamma(\tilde{\mathcal{CF}}_k) = k$.

(2) $k \in \text{Group A/Level } p, l \in \text{Group B/Level } q, q < p$. In this case, $\tilde{\mathcal{CF}}_k = \{(\hat{u}_{l-\alpha_{q+1}-\beta_q}, \hat{y}_1), (\hat{y}_1, u), \tilde{\mathcal{P}}_t, (y, \hat{u}_{l-\alpha_{q+1}-\beta_q})\} \cup \tilde{\mathcal{CF}}_d$ where $\tilde{\mathcal{P}}_t$ is some input-output path in $\tilde{\mathcal{D}}$, and $\tilde{\mathcal{CF}}_d$ is some d -cycle family in $\tilde{\mathcal{D}}$. As in case 1, if $\tilde{\mathcal{CF}}_d$ contains a d -edge which does not occur in $\tilde{\mathcal{CF}}_p$, then condition (iii)(b) is satisfied. On the other hand, if $t > p$, then $\tilde{\mathcal{P}}_t$ contains a d -edge which has no vertex in common with any $\tilde{\mathcal{CF}}_r, r \leq p$, and again condition (iii)(b) is satisfied. The only remaining possibility is the case when $\tilde{\mathcal{CF}}_d \subset \tilde{\mathcal{CF}}_p$ and $t \leq p$. This case, however, can be shown, as in case 1, to lead to a contradiction that $\gamma(\tilde{\mathcal{CF}}_k) < k$.

(3) $k \in \text{Group B/Level } p, l \in \text{Group B/Level } q, q \leq p$. $\tilde{\mathcal{CF}}_k$ is as in case 2. Again, if $\tilde{\mathcal{CF}}_d$ contains a d -edge which does not occur in $\tilde{\mathcal{CF}}_p$, or if $t \geq p + 2$, then condition (iii)(b) is satisfied. Otherwise, $\tilde{\mathcal{CF}}_d \subset \tilde{\mathcal{CF}}_p$ and $t \leq p + 1$, then $\gamma(\tilde{\mathcal{CF}}_k) < k$ unless $l = k$ and $\tilde{\mathcal{CF}}_d = \tilde{\mathcal{CF}}_p$, in which case $\tilde{\mathcal{CF}}_k = \tilde{\mathcal{CF}}_k^*$, and if $t \leq p$, then again $\gamma(\tilde{\mathcal{CF}}_k) < k$, both contradicting the assumption on $\tilde{\mathcal{CF}}_k$.

As a result, condition (iii)(b) is also satisfied, and the proof is complete for Case I.

Case II: The f -edge (y, u) is included in $\tilde{\mathcal{D}}$

In this case, $\tilde{\mathcal{CF}}$ is an f -cycle family of width n , which includes the f -cycle $\{(u, y), (y, u)\}$ with $\tilde{\mathcal{P}} = (u, y)$ being some input-output path in $\tilde{\mathcal{D}}$. Let the family of the remaining d -cycles of $\tilde{\mathcal{CF}}$ be denoted by \mathcal{CF} . Let $\{\tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2, \dots, \tilde{\mathcal{P}}_s\}, \tilde{\mathcal{P}}_s \neq \tilde{\mathcal{P}}$, be a minimal family of input-output paths in $\tilde{\mathcal{D}}$ such that any d -cycle in \mathcal{CF} has a vertex in common with some $\tilde{\mathcal{P}}_{j_s}$ and let $\tilde{\mathcal{P}}_{s+1} = \tilde{\mathcal{P}}$. We now define subfamilies of $\mathcal{CF}_j, 1 \leq j \leq s$, the same way as \mathcal{CF}_j 's are defined in Case I, but with respect to \mathcal{CF} and $\{\tilde{\mathcal{P}}_j\}$, rather than $\tilde{\mathcal{CF}}$ and $\{\tilde{\mathcal{P}}_j\}$, and similarly define integers α_j and $\beta_j, 1 \leq j \leq s$, in terms of \mathcal{CF}_j and $\tilde{\mathcal{P}}_j$. With these definitions, the proof follows the same lines as the proof of Case I, except that the integers at Level s are modified as shown in Table 2, where, obviously, $\alpha_{s+1} + \beta_s = n$. This completes the proof of Theorem 4. \square

Now several remarks are in order. The first remark is about our assumption that each cycle in the cycle family \mathcal{CF} of width n has a vertex in common with some input-output path. Obviously, this assumption is not essential for structural pole assignability of \mathcal{S} using a dynamic output feedback controller $\tilde{\mathcal{P}}$. However, it is needed for proving generic pole assignability using Theorem 2, and it has been observed by Şefik (1989) that it might be possible to remove this assumption by

modifying Theorem 2 to include more general cases when ψ_k of (13) contains linear terms in addition to a single quadratic term. The second remark is concerned with the restriction of \mathcal{S} to be a single-input/single-output system. Provided Theorem 2 is modified to remove the assumption mentioned above, this restriction can easily be relaxed. One way of doing this is to employ a preliminary constant output feedback to reduce the system to a single-input/single-output system without destroying structural controllability and observability, and then design $\hat{\mathcal{S}}$. A more efficient way, which also allows generic pole assignment using a smaller order dynamic compensator is to imitate the well-known results due to Brasch and Pearson (1970) and Ahmari and Vacroux (1973) in a structural setting. This, however, requires a structural interpretation of controllability and observability indices of \mathcal{S} , which is not a straightforward task.

5. Conclusion

Brasch and Pearson's (1970) renowned result which states that all the poles of a controllable and observable system can be assigned arbitrarily using a dynamic feedback compensator of order $L = \min \{L_c, L_o\} - 1$ where L_c and L_o are the controllability and observability indices of the system, is overly sufficient in most cases. This can be explained by the fact that their algebraic criterion does not take into account the structure of the system, which actually plays the most important role in the solvability of the problem.

In this paper we realize a qualitative investigation of the problem of arbitrary pole assignability, depending merely on the structural information about the system under consideration.

Our results detect a class of system structures for which constant output feedback is sufficient to place all the poles at desired locations. Many examples can be found for which, according to Brasch and Pearson's (1970) or Kimura's (1978) results, dynamic compensation is needed for pole assignment, while by our results constant output feedback is sufficient for the job.

Note that it may be possible to broaden the class of systems pole assignable by Theorem 2 by modifying the theorem somehow to include more general cases specified by Corollary 1. On the other hand, Corollary 1 is still a special case of some other result, namely Lemma 1, which probably contains hints on how to characterize a larger class of pole assignable structures.

Appendix

The choice algorithm

In this section we present an algorithm to check the existence of a set of n f -edges f_1, f_2, \dots, f_n in $\mathcal{D}(\mathcal{F})$ which satisfies conditions of Theorem 2, and to identify one such set if there exists any. The algorithm accepts as input

- I1: n , the number of state vertices in $\mathcal{D}(\mathcal{F})$;
- I2: $f = (f_1, f_2, \dots, f_v)$ a set of all f -edges, $v \geq n$;
- I3: for each $1 \leq k \leq n$, a list of all f -cycle families $\{\mathcal{CF}_{ks}\}$ of width k , each \mathcal{CF}_{ks} being specified as a product of the parametric weights of all the edges appearing in \mathcal{CF}_{ks} ,

and produces as output

- O1: a positive or negative response regarding the existence of a required set of f -edges, and if the response is positive;
- O2: the chosen subset $f_v = (f_1^*, f_2^*, \dots, f_n^*)$ of f (here we use a starred notation for the variable f -edges to distinguish between the orderings of f and f_v);
- O3: $\{\mathcal{CF}_k^*\}$, the list of particular simple f -cycle families defined by f_k^* , $1 \leq k \leq n$;
- O4: the fixed values (0 or 1) assigned to the f -edges in $f_c = f - f_v$.

The algorithm tries to construct an arborescence (a directed tree) $\mathcal{T} = (\mathcal{V}_c \cup \mathcal{V}_f, \mathcal{E}_t)$ having a longest path of length $2n$ by examining all cycle families \mathcal{CF}_{ks} , $k = 1, 2, \dots, n$, $s = 1, 2, \dots, n_k^c$; and all f -edges f_t^{ks} , $t = 1, 2, \dots, n_k^c$, appearing in each \mathcal{CF}_{ks} . It halts with a positive response as soon as such a longest path is constructed, and with a negative response if no such path can be formed. The vertices of \mathcal{T} are arranged in $n + 1$ levels, each of which, except level 0, is further divided into two sublevels. The vertices at the first sublevels constitute \mathcal{V}_c , and are called the c -vertices, while the vertices at the second sublevels constitute \mathcal{V}_f , and are called the f -vertices. Each c -vertex at level k is the son of some f -vertex at level $k - 1$, and corresponds to an f -cycle family \mathcal{CF}_{ks} of width k , while each f -vertex at level k is the son of some c -vertex \mathcal{CF}_{ks} at the same level, and corresponds to an f -edge that occur in \mathcal{CF}_{ks} , $1 \leq k \leq n$. Level 0 contains a single f -vertex, denoted by f_0^* , which is the root of \mathcal{T} . The algorithm proceeds as follows.

Suppose that a path \mathcal{P}_{k-1} of length $2(k - 1)$ is constructed from f_0^* to some f vertex f_{k-1}^* at level $(k - 1)$, with some f -edges of $\mathcal{D}(\mathcal{T})$ assigned to the branches and f -vertices on \mathcal{P}_{k-1} are assigned to the f vertices of \mathcal{P}_{k-1} . If no such \mathcal{CF}_{ks} exists, terminate the path \mathcal{P}_{k-1} , and search for an unexplored f -vertex at level $(k - 1)$ to replace f_{k-1}^* . If there exist one or more such cycle families, construct a c -vertex for each of them and extend a branch from f_{k-1}^* to these c -vertices. Pick any one of these c -vertices, say \mathcal{CF}_{ks} , $s = 1, 2, \dots, n_k^c$, and label it as \mathcal{CF}_k^* . Corresponding to each f -edge that occurs in $\mathcal{CF}_k^* = \mathcal{CF}_{ks}$ construct an f -vertex, f_t^{ks} , $t = 1, 2, \dots, n_k^c$, extend a branch from \mathcal{CF}_k^* to each f_t^{ks} , and assign all other f -edges in \mathcal{CF}_k^* to the branch $(\mathcal{CF}_k^*, f_t^{ks})$ of \mathcal{T} . Pick one of the f -vertices, say f_t^{ks} , and check if the assignment $f_k^* = f_t^{ks}$ violates the conditions of Theorem 2. If not, set $f_k^* = f_t^{ks}$, and repeat the whole procedure with $k - 1$, f_{k-1}^* and \mathcal{P}_{k-1} replaced by k , f_k^* and \mathcal{P}_k . If the assignment $f_k^* = f_t^{ks}$ violates the conditions of Theorem 2, terminate the path from f_0^* to f_t^{ks} , and pick another unexplored f -vertex to replace f_t^{ks} . If none of f_t^{ks} can be chosen as f_k^* , go back to the upper sublevel to replace \mathcal{CF}_k^* with another unexplored c -vertex \mathcal{CF}_{ks} . If all the paths through all \mathcal{CF}_{ks} are terminated, search for an unexplored f -vertex at level $(k - 1)$ to replace f_{k-1}^* . In checking whether the assignment $f_k^* = f_t^{ks}$ violates the conditions of Theorem 2, the f -edges of $\mathcal{D}(\mathcal{T})$ that are assigned to any branch of \mathcal{P}_k^* are assumed to be fixed at some non-zero value (at 1, for convenience), and all f -edges other than these and f_j^* , $1 \leq j \leq k$ can be fixed or variable, as appropriate.

With this introduction, we now state the choice algorithm to identify $f_v = (f_1^*, f_2^*, \dots, f_n^*)$, where we adopt the following notation:

- k index to scan the levels of \mathcal{T} , $0 \leq k \leq n$
- n_k^c number of distinct f -cycle families of width k in $\mathcal{D}(\mathcal{T})$
- s_k index to scan the c -vertices of \mathcal{T} at level k , $0 \leq s_k \leq n_k^c$
- \mathcal{CF}_k^* the c -vertex chosen at level k

- n_k^* number of distinct f -edges of $\mathcal{D}(\mathcal{F})$ that appear in \mathcal{CF}_k^*
- t_k index to scan the f -vertices of \mathcal{T} at level k , $0 \leq t_k \leq n_k^*$
- f_k^* the f -vertex chosen at level k

The algorithm

- Step 1. Set $k \leftarrow 1$, and construct vertex f_0^* of \mathcal{T}
- Step 2. Add the c -vertices \mathcal{CF}_{ks} and the branches $(f_{k-1}^*, \mathcal{CF}_{ks})$ to \mathcal{T} , $1 \leq s \leq n_k^c$, and set $s_k \leftarrow 0$.
- Step 3. Set $s_k \leftarrow s_k + 1$. If $s_k \leq n_k^c$ go to Step 5.
- Step 4. Set $k \leftarrow k - 1$. If $k = 0$, stop. No choice of f_v is possible. Otherwise, go to Step 7.
- Step 5. If \mathcal{CF}_{ks_k} contains an f -edge corresponding to an f -vertex f_j^* of \mathcal{T} , $i \leq j \leq k - 1$, terminate the path from f_0^* to \mathcal{CF}_{ks_k} , and go to Step 3. Otherwise, let $\mathcal{CF}_k^* = \mathcal{CF}_{ks_k}$.
- Step 6. Add the f -vertices f_{kt} and the branches $(\mathcal{CF}_k^*, f_{kt})$ to \mathcal{T} , $1 \leq t \leq n_n^*$, and set $t_k \leftarrow 0$.
- Step 7. Set $t_k \leftarrow t_k + 1$. If $t_k > n_k^*$, go to Step 3.
- Step 8. If k_{kt_k} is assigned to any branch $(\mathcal{CF}_j^*, f_j^*)$ of \mathcal{T} , $1 \leq j \leq k - 1$, terminate the path from f_0^* to f_{kt_k} , and go to step 7. Otherwise, assign all the f -edges in \mathcal{CF}_k^* , except f_{kt_k} , to the branch $(\mathcal{CF}_k^*, f_{kt_k})$.
- Step 9. Delete all f -edges of $\mathcal{D}(\mathcal{F})$ except f_{kt_k} , those that correspond to the f -vertices f_j^* , $1 \leq j \leq k - 1$, and those that are assigned to the branches $(\mathcal{CF}_j^*, f_j^*)$, $1 \leq j \leq k - 1$. Convert all f -edges that are assigned to the branches $(\mathcal{CF}_j^*, f_j^*)$ to d -edges by choosing their weights to be unity. If $f_1^*, f_2^*, \dots, f_{k-1}^*$ and f_{kt_k} do not satisfy the conditions of Theorem 2 for the remaining subgraph and with n replaced by k , terminate the path from f_0^* to f_{kt_k} and go to Step 7. Otherwise, let $f_k^* = f_{kt_k}$.
- Step 10. If $k < n$, set $k \leftarrow k + 1$, and go to Step 2.
- Step 11. Let $f_v = (f_1^*, f_2^*, \dots, f_n^*)$. Convert all the remaining f -edges of $\mathcal{D}(\mathcal{F})$ into d -edges by fixing their weights to 1 if they are assigned to some branch $(\mathcal{CF}_k^*, f_k^*)$ of \mathcal{T} , $1 \leq k \leq n$, and to 0 otherwise. Stop.

Example 1

Consider the digraph $\mathcal{D}(\mathcal{F})$ of Fig. 6, in which units weight is assigned to any d -edge adjacent from an input or to an output vertex. In this example, this causes no loss of generality as every input or output vertex has a unique edge adjacent from or to itself. We want to identify an $f_v = (f_1^*, f_2^*, f_3^*, f_4^*)$, if any exists. We have $n = 4$ and $f = (f_1, f_2, f_3, f_4, f_5, f_6)$, and the list of all f -cycle families $\{\mathcal{CF}_{ks}\}$ of width k , $1 \leq k \leq n$, is given in Table 3.

Figure 7 shows the arborescence constructed during the application of the choice algorithm. The algorithm stops with a positive response identifying $f_v = (f_6, f_3, f_1, f_2)$ while fixing $f_4 = f_5 = 0$. The corresponding choice of particular

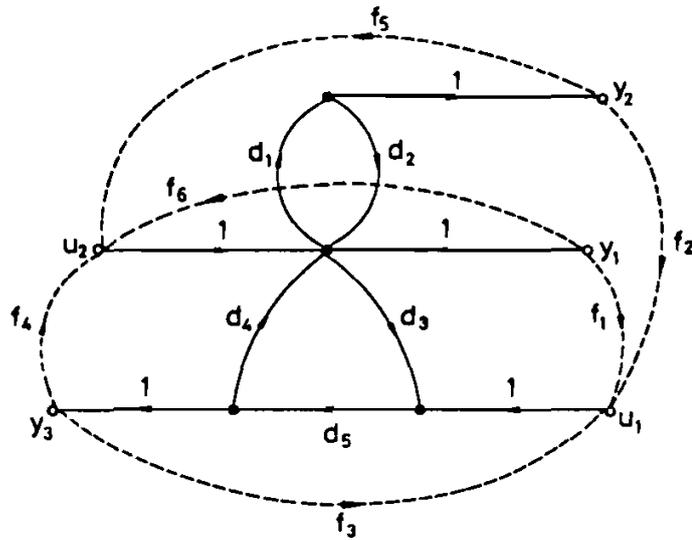


Figure 6. $\mathcal{D}(\mathcal{F})$ of Example 1.

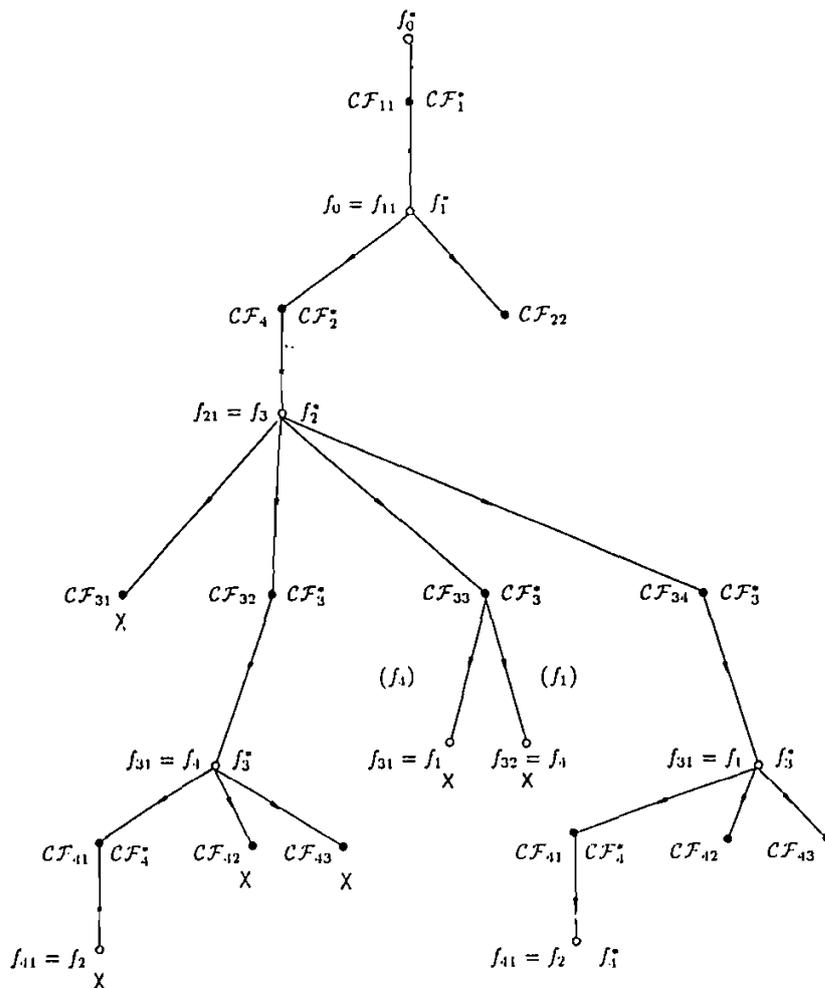


Figure 7. Arborescence generated by the chosen algorithm in Example 1.

| k | \mathcal{CF}_{k_s} |
|-----|--|
| 1 | $\{f_6\}$ |
| 2 | $\{d_5, f_3\}$ $\{d_1, f_5\}$ |
| 3 | $\{\{d_5, f_3\}, \{f_6\}\}$ $\{d_5, d_3, f_4\}$ $\{d_5, f_1, f_4\}$ $\{d_4, d_5, f_1\}$ |
| 4 | $\{d_1, d_4, d_5, f_2\}$ $\{d_1, f_4, d_5, f_2\}$ $\{\{d_5, f_3\}, \{d_1, f_5\}\}$ |

Table 3. f -Cycle families in $\mathcal{D}(\mathcal{F})$ of Fig. 6.

simple f -cycle families is $\mathcal{CF}_1^* = \{f_6\}$, $\mathcal{CF}_2^* = \{d_5, f_3\}$, $\mathcal{CF}_3^* = \{d_4, d_5, f_1\}$, $\mathcal{CF}_4^* = \{d_1, d_4, d_5, f_2\}$.

This example illustrates the significance of Theorem 1 and Theorem 2 and hence the usefulness of our choice algorithm: in a classical approach, in order to place all the poles of a system possessing a digraph $\mathcal{D}(\mathcal{F})$, as given in Fig. 6, at desired locations, one would attempt to use a dynamic output compensator, whereas we showed above that constant output feedback was sufficient for the job. \square

REFERENCES

- AHMARI, R., and VACROUX, A. G., 1973, On pole assignment in linear systems with fixed order compensators. *International Journal of Control*, **17**, 397–404.
- BRASCH, F. M., JR, and PEARSON, J. B., 1970, Pole placement using dynamic compensation. *I.E.E.E. Transactions on Automatic Control*, **15**, 34–43.
- DAVISON, E. J., and CHATTERJEE, R., 1971, A note on pole assignment in linear systems with incomplete state feedback. *I.E.E.E. Transactions on Automatic Control*, **16**, 98–99.
- DAVISON, E. J., and WANG, S. H., 1973, Properties of linear time-invariant systems subject to arbitrary output and state feedback. *I.E.E.E. Transactions on Automatic Control*, **18**, 24–34.
- HARARY, F., NORMAN, R. Z., and CARTWRIGHT, D., 1965, *Structural Models: An Introduction to the Theory of Directed Graphs* (New York: Wiley).
- KIMURA, H., 1978, On pole assignment by output feedback. *International Journal of Control*, **28**, 11–22.
- LIN, C. T., 1974, Structural controllability. *I.E.E.E. Transactions on Automatic Control*, **19**, 201–208.
- PICHAJ, V., SEZER, M. E., and ŠILJAK, D. D., 1984, A graph-theoretic characterization of structurally fixed modes. *Automatica*, **20**, 247–250.
- REINSCHKE, S. J., 1984, Graph-theoretic characterization of fixed modes in centralized and decentralized control. *International Journal of Control*, **39**, 715–729; 1987, Algebraic formulation of arbitrary pole-assignability by static output feedback. *Ibid.* **46**, 1947–1977.
- SAGE, A. P., 1977, *Methodology for Large Scale Systems* (New York: McGraw-Hill).
- ŞEFİK, A., 1989, structural analysis of pole assignment and stabilization in dynamic systems. Ph.D. thesis, Bilkent University, Ankara.
- SEZER, M. E., and ŠILJAK, D. D., 1981, On structurally fixed modes. *Proceedings of the I.E.E.E. Symposium on Circuits and Systems*, Chicago.
- SHIELDS, R. W., and PEARSON, J. B., Structural controllability of multi-input linear systems. *I.E.E.E. Transactions on Automatic Control*, **21**, 203–212.
- TARJAN, R., 1972, Depth-first search and linear graph algorithms. *SIAM Journal of Computing*, **1**, 146–160.