2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Complexity of Checking the Existence of a Stabilizing Decentralized Controller

Javad Lavaei and Somayeh Sojoudi

Abstract— **Given an interconnected system, this paper is concerned with the time complexity of verifying if any given unrepeated mode of the system is a decentralized fixed mode (DFM). It is shown that checking the decentralized fixedness of any distinct mode is tantamount to testing the strong connectivity of a digraph formed based on the system. It is subsequently proved that the time complexity of this decision problem using the proposed approach is the same as the complexity of matrix multiplication. This work concludes that the identification of distinct decentralized fixed modes (by means of a deterministic algorithm, rather than a randomized one) is computationally very easy, although the existing algorithms for solving this problem would wrongly imply that it is cumbersome. This paper provides not only a complexity analysis, but also an efficient algorithm for tackling the underlying problem.**

I. Introduction

An interconnected system consists of a number of interacting subsystems, which could be homogeneous or heterogeneous. It is evident that many real-world systems can be modeled as interconnected systems, some of which are communication networks, large space structures, power systems, and chemical processes [1], [2], [3], [4], [5]. The classical control techniques often fail to control such systems, in light of some well-known practical issues such as computation or communication constraints. This has given rise to the emergence of the decentralized control area which aims to design non-classical structurally constrained controllers [6]. More precisely, a (conventional) decentralized controller comprises a set of non-interacting local controllers corresponding to different subsystems.

The notion of decentralized fixed modes (DFM) was introduced in [7] to characterize those modes of the system which cannot be moved using a linear time-invariant (LTI) decentralized controller. Several methods have been proposed in the literature to find the DFMs of a system [8], [9], [10], [11]. For instance, an algebraic characterization of DFMs was provided in [8]. The method given in [9], on the other hand, characterizes the DFMs of a system in terms of its transfer function. It was also shown in [12] that the DFMs of any system can be found by checking the transmission zeros of a set of artificial systems derived from the original system. In [10], an algorithm was presented to identify the DFMs of the system by computing the rank of a set of matrices. Unfortunately, the number of the systems whose transmission zeros need to be checked in [12] and the number of matrices whose ranks are to be computed in [10] depend exponentially on the number of subsystems of the original system.

On the other hand, a method is delineated in [13] stating that in order to numerically find the DFMs of a system, it is sufficient to apply a randomly generated static decentralized controller to the system, and then verify what modes of the system are still fixed. Nevertheless, this method is often inaccurate for large-scale systems. More precisely, calculating the eigenvalues of a large-size matrix is normally associated with some errors (especially when the matrix possesses complex eigenvalues), which makes it impossible to distinguish the fixed modes from the approximate fixed modes [14]. Another issue is that a generic static decentralized controller may not be able to sufficiently displace a mode so that it is recognized as a non DFM.

A graph-theoretic method was proposed in the recent work [11], which constructs a bipartite graph corresponding to each unrepeated mode of the system. This work states that the mode is a DFM if and only if the graph contains a bipartite subgraph satisfying two specific properties. Although this method turns out to be extremely simpler than other available methods, it is not clear how to systematically verify the existence of such a subgraph.

Consider a decision problem, whose answer to be found is “yes” or “no”. An algorithm provided for this problem is efficient if its time complexity is satisfactory. Informally speaking, the time complexity measures the number of machine instructions executed during the running time of the algorithm (as a function of the size of the input). It is well-understood in computer science that if there exists an efficient randomized algorithm for a decision problem, normally there should also exist a deterministic algorithm with similar complexity. In other words, randomized algorithms cannot be far more efficient than deterministic algorithms. Regarding the decentralized question posed here (i.e. finding the DFMs of a system), the work [13] shows that there exists an efficient randomized algorithm, whereas the available deterministic algorithms have high time complexities. Based on the above-mentioned discussion, one would conjecture that there exists an efficient deterministic algorithm for the underlying decentralized problem. Finding such an algorithm and investigating its properties are central to the current work.

Given an LTI interconnected system realized in the canonical form, consider a distinct mode of the system. The primary objective of this paper is to determine the time complexity of deciding whether this mode is a DFM of the system. To tackle this decision problem, a digraph is constructed by means of an algorithm, whose time complexity is the same as the complexity of matrix multiplication. It is then shown that the answer to the posed decision problem is affirmative

This work has been supported by AFOSR and Air Force MURI.

The authors are with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, USA (emails: lavaei@cds.caltech.edu; sojoudi@cds.caltech.edu).

978-1-4244-4524-0/09/$25.00 ©2009 AACC

if and only if the digraph is not strongly connected. The time complexity of the latter problem (checking the strong connectivity) is quadratic with respect to the number of subsystems of the system. It is eventually concluded that the time complexity of the original decision problem is the same as that of matrix multiplication, being at most equal to $O(n^{2.376})$, where n denotes the order of the given system. Note that it is extremely unlikely to find another algorithm for this decentralized problem which uses only matrix operations with (complexity) exponents lower than that of matrix multiplication (i.e. the ordinary operations *multiplication, inversion, determinant, rank,* cannot be used in that algorithm). This signifies that the obtained complexity order is believed to be the best possible one.

The paper is organized as follows. The problem is formulated in Section II, where some preliminaries are provided. The main results are developed in Section III, followed by a numerical example in Section IV. Finally, some concluding remarks are drawn in Section V.

II. Preliminaries and problem formulation

Consider an LTI interconnected system $\mathcal{S}$ consisting of ν subsystems $S_1, S_2, ..., S_\nu$, represented by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + \sum_{j=1}^{\nu} B_j u_j(t) \\ y_i(t) &= C_i x(t) + \sum_{j=1}^{\nu} D_{ij} u_j(t), \ i \in \boldsymbol{\nu} := \{1, 2, ..., \nu\} \end{aligned} \tag{1}$$

where $x(t) \in \Re^n$ is the state, and $u_i(t) \in \Re^{m_i}$ and $y_i(t) \in \Re^{r_i}$, $i \in \boldsymbol{\nu}$, are the input and the output of the i^{th} subsystem, respectively. Define now:

$$\begin{aligned} B &:= \left[\begin{array}{ccc} B_1 & \cdots & B_\nu \end{array} \right], \\ C &:= \left[\begin{array}{ccc} C_1^T & \cdots & C_\nu^T \end{array} \right]^T, \\ D &:= \left[\begin{array}{ccc} D_{11} & \cdots & D_{1\nu} \\ \vdots & \ddots & \vdots \\ D_{\nu 1} & \cdots & D_{\nu\nu} \end{array} \right], \\ m &:= \sum_{i=1}^{\nu} m_i, \quad r := \sum_{i=1}^{\nu} r_i \end{aligned} \tag{2}$$

A (conventional) decentralized controller for the system $\mathcal{S}$ is composed of a set of ν local controllers, where the i^{th} local controller, $i \in \boldsymbol{\nu}$, observes only the local output $y_i(t)$ to construct the local input $u_i(t)$ of the i^{th} subsystem. The following definition was presented in [7] for strictly proper systems and generalized in [10] to proper systems.

Definition 1: A mode σ is said to be a decentralized fixed mode (DFM) of the system $\mathcal{S}$ if there exists no static decentralized controller to displace this mode. In other words, σ is a DFM of the system $\mathcal{S}$ if the relation:

$$\sigma \in \text{sp}\left(A + BK(I - DK)^{-1}C\right) \tag{3}$$

holds for every block diagonal matrix K whose i^{th} block entry, $i \in \boldsymbol{\nu}$, is a matrix of dimension $m_i \times r_i$, where sp$(\cdot)$ stands for the matrix spectrum.

It is noteworthy that as shown in [10], a DFM is fixed with respect to not only static decentralized controllers but also all types of LTI decentralized controllers. In what follows, different methods for finding the DFMs of a system are outlined.

A. Matrix rank checking

According to [10], a mode σ is a DFM of the system $\mathcal{S}$ if and only if there exist a permutation of $\{1, 2, ..., \nu\}$ denoted by $(i_1, i_2, ..., i_\nu)$ and an integer $p \in [0, \nu]$ such that the rank of the following matrix is less than n:

$$\left[\begin{array}{ccccc} A - \sigma I_n & B_{i_1} & B_{i_2} & \cdots & B_{i_p} \\ C_{i_{p+1}} & D_{i_{p+1}i_1} & D_{i_{p+1}i_2} & \cdots & D_{i_{p+1}i_p} \\ C_{i_{p+2}} & D_{i_{p+2}i_1} & D_{i_{p+2}i_2} & \cdots & D_{i_{p+2}i_p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_{i_\nu} & D_{i_\nu i_1} & D_{i_\nu i_2} & \cdots & D_{i_\nu i_p} \end{array} \right] \tag{4}$$

This clearly signifies that computing the DFMs of the system $\mathcal{S}$ using this method is cumbersome, due to the necessity of checking the rank of 2^ν matrices, in general.

B. Randomized algorithm

Pick a matrix $K \in \Re^{r \times m}$ at random and consider the matrices A and $A + BK(I - DK)^{-1}C$. The works [13] and [10] state that the DFMs of the system are, almost surely, the common eigenvalues of these two matrices. This gives rise to a randomized algorithm that almost always works correctly. As explained in the introduction, this method suffers from some numerical issues. Nonetheless, this technique indicates that there is a simple randomized algorithm for finding DFMs, whose complexity is much lower than the deterministic one explained above (i.e. testing the rank of an exponential number of matrices).

C. Derandomization

The work [10] proposes a derandomization technique for the randomized algorithm given in [13] (discussed above). Observe that a decentralized gain matrix K has $\sum_{i=1}^{\nu} m_i r_i$ free parameters, sitting on the block diagonal of this matrix. For every natural number j satisfying the inequality $j \leq \sum_{i=1}^{\nu} m_i r_i$, pick j of these free parameters, give arbitrary nonzero values to them, and set the remaining free parameters to zero. This leads to a structured decentralized gain matrix. Repeating this procedure for all possible combinations yields p block-diagonal matrices $K_1, K_2, ..., K_p$, where:

$$p = 2^{\sum_{i=1}^{\nu} m_i r_i} - 1 \tag{5}$$

The derandomized algorithm says that the DFMs of the system are the common eigenvalues of the matrices $A + BK_i(I - DK_i)^{-1}C, \ i = 1, 2, ..., p$. Note that although the randomized algorithm mentioned in the preceding subsection runs in polynomial time, its derandomized counterpart runs in exponential time.

D. Graph-theoretic approach

Assume that σ is an eigenvalue of A with multiplicity 1, which is also an observable and controllable mode of the system $\mathcal{S}$. With no loss of generality, suppose that the matrix A is in the following canonical form:

$$A = \begin{bmatrix} \sigma & 0 \\ 0 & \mathbf{A} \end{bmatrix} \tag{6}$$

where $\mathbf{A}$ is a matrix of appropriate dimension (this can be achieved by using a proper similarity transformation, if need be). Define now:

$$M(\sigma) := \mathbf{C}(\mathbf{A} - \sigma I_{n-1})^{-1}\mathbf{B} - D \tag{7}$$

where:

- $\mathbf{C}$ is derived from C by eliminating its first column.
- $\mathbf{B}$ is obtained from B by removing its first row.

Denote the (i,j) block entry of $M(\sigma)$ with $M_{ij}(\sigma) \in \Re^{r_i \times m_j}$, for every $i,j \in \boldsymbol{\nu}$.

Definition 2: Let $\mathcal{G}(\sigma)$ be a bipartite graph constructed as follows:

- Consider two sets of vertices, namely set 1 and set 2, with ν vertices in each of them.
- For every $i,j \in \boldsymbol{\nu},\ i \neq j$, connect vertex i in set 1 to vertex j in set 2 if all of the following conditions are satisfied:
  - The first column of C_i is zero.
  - The first row of B_j is zero.
  - $M_{ij}(\sigma)$ is a zero matrix

We proposed the next result in [11] to verify whether or not σ is a DFM of the system $\mathcal{S}$.

Theorem 1: The mode σ is a DFM of the system $\mathcal{S}$ if and only if the graph $\mathcal{G}(\sigma)$ contains a subgraph $\mathcal{G}_o(\sigma)$ with the following properties:

- It is complete bipartite.
- If $(i_1, i_2, ..., i_p)$ and $(j_1, j_2, ..., j_q)$ represent the sets of vertices of $\mathcal{G}_o(\sigma)$ (i.e. set 1 and set 2 of $\mathcal{G}_o(\sigma)$), then $(i_1, ..., i_p, j_1, ..., j_q)$ is a permutation of the set $\boldsymbol{\nu}$.

Although this method seems to be far simpler than the deterministic methods outlined above, it is not clear how to verify the existence of such a subgraph $\mathcal{G}_o(\sigma)$ systematically.

E. Objective of this work

This work develops the result of [11] under the assumption that the matrix A is in the canonical form (6). The objective is twofold. First, it is desired to propose a simple deterministic algorithm to check whether σ is a DFM of the system $\mathcal{S}$. Second, it is aimed to figure out the time complexity of this problem using deterministic algorithms.

III. Main results

Assume that the quantities m, r, ν are all less than or equal to n. This realistic assumption is made so that the complexity of computing the DFMs of $\mathcal{S}$ can be written only in terms of n. The following definitions turn out to be convenient in proceeding with the development of the paper.

Definition 3: Define $\tilde{\mathcal{G}}(\sigma)$ to be a directed graph (digraph) constructed as follows:

- Consider ν vertices, labeled as $1, 2, ..., \nu$.
- For every $i,j \in \boldsymbol{\nu},\ i \neq j$, connect vertex i to vertex j by means of a directed edge if any of the conditions given below is satisfied:
  - The first column of C_i is a nonzero vector.
  - The first row of B_j is a nonzero vector.
  - $M_{ij}(\sigma)$ is not a zero matrix.

Definition 4: The digraph $\tilde{\mathcal{G}}(\sigma)$ is said to be strongly connected if there exists a directed path from vertex v_1 to vertex v_2, for every disparate vertices v_1 and v_2 of the digraph.

It is well-known from graph theory that $\tilde{\mathcal{G}}(\sigma)$ can be uniquely decomposed as a union of strongly connected components, namely $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_k$, such that:

- $\mathcal{C}_i,\ i = 1, 2, ..., k$, is a strongly connected induced subgraph of $\tilde{\mathcal{G}}(\sigma)$.
- For every $i,j \in \{1, 2, ..., k\},\ i < j$, there is no directed edge going from $\mathcal{C}_j$ to $\mathcal{C}_i$.

This fact will be exploited in the sequel to present one of the main results of the paper.

Theorem 2: The mode σ is a DFM of the system $\mathcal{S}$ if and only if the digraph $\tilde{\mathcal{G}}(\sigma)$ is not strongly connected.

Proof of sufficiency: Assume that the digraph $\tilde{\mathcal{G}}(\sigma)$ is not strongly connected. In light of the discussion given prior to this theorem, the set $\{1, 2, ..., \nu\}$ can be partitioned as $\{i_1, i_2, ..., i_p\}$ and $\{j_1, j_2, ..., j_q\}$ such that there exits no directed edge from vertex i_α to vertex j_β in the digraph $\tilde{\mathcal{G}}(\sigma)$, for all $\alpha \in \{1, 2, ..., p\}$ and $\beta \in \{1, 2, ..., q\}$. Consider vertices $i_1, i_2, ..., i_p$ in set 1 and vertices $j_1, j_2, ..., j_q$ in set 2 of the bipartite graph $\mathcal{G}(\sigma)$. Denote with $\mathcal{G}_o(\sigma)$ the bipartite subgraph induced by these vertices. It is straightforward to observe that this subgraph is complete bipartite (using Definitions 2 and 3). On the other hand:

$$\{1, 2, ..., \nu\} = \{i_1, i_2, ..., i_p\} \cup \{j_1, j_2, ..., j_q\} \tag{8}$$

Now, it follows immediately from Theorem 1 that σ is a DFM.

Proof of necessity: Assume that σ is a DFM. It is to be proved that the digraph $\tilde{\mathcal{G}}(\sigma)$ is not strongly connected. To this end, one can utilize Theorem 1 to deduce that the graph $\mathcal{G}(\sigma)$ possesses a bipartite subgraph $\mathcal{G}_o(\sigma)$ with the two properties mentioned earlier. Denote the sets of vertices of this bipartite subgraph with $\{i_1, i_2, ..., i_p\}$ and $\{j_1, j_2, ..., j_q\}$. Due to the properties of $\mathcal{G}_o(\sigma)$, not only is the relation (8) satisfied, but the following are true for every $\alpha \in \{1, 2, ..., p\}$ and $\beta \in \{1, 2, ..., q\}$:

- The first column of C_α is zero.
- The first row of B_β is zero.
- $M_{\alpha\beta}(\sigma)$ is a zero matrix.

This, together with the relation (8), means that the two subgraphs of $\tilde{\mathcal{G}}(\sigma)$ induced by the respective sets of vertices $\{i_1, i_2, ..., i_p\}$ and $\{j_1, j_2, ..., j_q\}$ cover all vertices of the digraph $\tilde{\mathcal{G}}(\sigma)$ and, besides, there is no directed edge from the subgraph induced by $\{i_1, i_2, ..., i_p\}$ to the other one. Hence, the digraph $\tilde{\mathcal{G}}(\sigma)$ is not strongly connected. ■

Theorem 2 states that checking the decentralized fixedness of σ reduces to testing the strong connectivity of the digraph $\tilde{\mathcal{G}}(\sigma)$. Fortunately, the latter problem is a very simple combinatorial problem, for which several methods have been developed. For instance, one can use Kosaraju's algorithm, which has been regarded as the simplest method for this graph problem [15]. This algorithm performs two complete traversals of the graph and the idea behind it is a depth-first search. Alternatively, Tarjan's algorithm can be employed, whose efficiency is better than Kosaraju's algorithm [16]. Another efficient algorithm, which is mostly suitable for dense graphs, is the Cheriyan/Mehlhorn/Gabow algorithm [17]. Let Tarjan's algorithm be adopted in this paper to check the strong connectivity of $\tilde{\mathcal{G}}(\sigma)$. Note that this algorithm has been implemented in the Bioinformatics Toolbox of MATLAB.

Theorem 3: Consider the decision problem "whether or not the mode σ is a DFM". Let the time complexity of matrix multiplication in $\Re^{n\times n}$ be denoted by $O(n^{\omega})$, where ω is a positive real.

i) The posed decision problem can be solved in $O(n^{\omega})$ time by computing the matrix $M(\sigma)$, and then testing the strong connectivity of its associated digraph $\tilde{\mathcal{G}}(\sigma)$.

ii) If there exists another algorithm for this decision problem which runs in $O(n^{\bar{\omega}})$ where $\bar{\omega} < \omega$, the algorithm must not use any of the following operations over arbitrary unstructured matrices of approximate dimension $n \times n$: *matrix multiplication, matrix inversion, determinant, LUP-decomposition, computing the characteristic polynomial, orthogonal basis transformation, matrix rank.*

Proof of Part (i): Denote the number of edges of $\tilde{\mathcal{G}}(\sigma)$ with η. It is well-known that Tarjan's algorithm runs in $O(\nu+\eta)$ time in order to test the strong connectivity of $\tilde{\mathcal{G}}(\sigma)$. Since η is less than or equal to $\nu(\nu-1)$, the complexity of checking the connectivity of the graph $\tilde{\mathcal{G}}(\sigma)$ is at most $O(\nu^2)$. On the other hand, it is known that matrix inversion and matrix multiplication have the same time complexity exponent [18]. Since $M(\sigma)$ is computed by one matrix inversion and two matrix multiplications, $M(\sigma)$ can be found in $O(n^{\omega})$ time. Moreover, the complexity of matrix multiplication over $\Re^{n\times n}$ is at least $O(n^2)$, because there are n^2 entries in the matrix which must be part of any computation. Hence, the quantity ω is at least equal to 2. These results lead to the conclusion that checking the decentralized fixedness of σ can be accomplished in $O(n^{\omega})+O(\nu^2) = O(n^{\omega})$ time (note that $\nu \leq n$, by assumption).

Proof of Part (ii): The proof of this part follows from part (i) and the fact that the operations pointed out in the theorem have (complexity) exponents greater than or equal to that of matrix multiplication [18]. ■

Remark 1: If the standard method of matrix multiplication is used to compute $M(\sigma)$, the time complexity of checking the decentralized fixedness of σ turns out to be $O(n^3)$, in light of Theorem 3. However, one can employ Coppersmith-Winograd algorithm for matrix multiplication to reduce this complexity to $O(n^{2.376})$ [19].

IV. Numerical example

Let $\mathcal{S}$ be composed of 10 single-input single-output interconnected subsystems, with the state-space matrices given in (9). Notice that the matrix A is already in the canonical form (6) for $\sigma = 1$.

It is desired now to check whether the mode $\sigma = 1$ is a DFM of the system. To this end, the matrix $M(\sigma)$ introduced in (7) should be computed first. The digraph $\tilde{\mathcal{G}}(\sigma)$ constructed in terms of this matrix is depicted in Figure 1. This graph has 10 vertices and 47 edges. Tarjan's Algorithm can be employed to traverse all these edges and vertices in order to find the strongly connected components of this graph. This is carried out in the Bioinformatics toolbox of MATLAB using the command "graphconncomp". Vertices 1, 2 and 3 in Figure 1 have been colored dark blue, meaning that MATLAB has detected them as a connected component of the digraph and the remaining vertices as another component. Consequently, the digraph is not strongly connected, and hence $\sigma = 1$ is a DFM. Note that even though the graph $\tilde{\mathcal{G}}(\sigma)$ seems to be complex, its connectivity verification is an easy job which can be accomplished in quadratic running time (in terms of ν). In this regard, it is worth mentioning that computing the matrix $M(\sigma)$ is more involved than testing the connectivity of $\tilde{\mathcal{G}}(\sigma)$. This clearly shows the simplicity of the method proposed here.

V. Conclusions

This paper deals with the time complexity of checking the existence of a stabilizing linear time-invariant (LTI) decentralized controller for a given LTI interconnected system. In particular, the complexity of computing the decentralized fixed modes (DFMs) of a system is studied. It is well-known that the existing deterministic methods for this problem are computationally intractable, whereas the available randomized numerical method is fairly simple. The objective is to determine the true complexity of solving this problem using a deterministic algorithm. To this end, it is shown that checking whether a certain (unrepeated) mode of the system is a DFM amounts to testing the strong connectivity of some digraph. This gives rise to proving that the verification of the decentralized fixedness of a distinct mode of the system has the same time complexity as matrix multiplication and inversion.

References

[1] D. M. Stipanovic, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," *Automatica*, vol. 40, no. 8, pp. 1285-1296, 2004.

[2] A. I. Zecevic, G. Neskovic, and D. D. Siljak, "Robust decentralized exciter control with linear feedback," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1096-1103, 2004.

[3] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465- 1476, 2004.

[4] J. Lavaei, A. Momeni, and A. G. Aghdam, "Spacecraft formation control in deep space with reduced communication requirement", *IEEE Transactions on Control System Technology*, vol. 16, no. 2, pp. 268-278, 2008.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 1 & 1 & 0 & 1 & 0 & -1 & 1 \\ 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 1 \\ -1 & 1 & 0 & 0 & -1 & -1 & 1 & 0 & -1 & 1 \\ 1 & -1 & -1 & 0 & 0 & 1 & 0 & -1 & 1 & 1 \\ 0 & 1 & -1 & 0 & -1 & -1 & 1 & 1 & 1 & -1 \\ -1 & 0 & 1 & -1 & 1 & -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \\ -1 & 1 & 1 & -1 & 0 & 0 & -1 & 1 & 1 & 0 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & 0 & -1 & -1 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & -1 & 1 & 1 & -1 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 & 0 & -1 & 0 \\ 0 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & -1 & 1 & -1 & -1 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 & -1 & 1 & 0 & -1 \end{bmatrix}, \tag{9}$$

$$D = \begin{bmatrix} 1 & -31/3 & 1 & 22 & 125/6 & 86/3 & 43/3 & -7 & -5 & -107/6 \\ 1 & 1 & 0 & 3 & 10/3 & 23/3 & 13/3 & -2 & -3 & -10/3 \\ 1 & 1 & 1 & 2 & 10/3 & -7/3 & -8/3 & 2 & 4 & -4/3 \\ 1 & 5/3 & 1 & 1 & 4/3 & -7/3 & 1 & 3 & 1 & 5/3 \\ 1 & 1 & 1 & 8 & 1 & 28/3 & 1 & 1 & 1 & 1 \\ -1/2 & 1 & 1 & 1 & 9/2 & 2 & 1 & -1 & 2 & -3/2 \\ -2 & 20/3 & 1 & 1 & -41/3 & 1 & 1 & 4 & 1 & 1 \\ 1 & 1 & 0 & 12 & 1 & 17 & 9 & -5 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -25/2 \\ 1 & 1 & -2 & 12 & 1 & 49/3 & 26/3 & -5 & 1 & -29/3 \end{bmatrix}$$

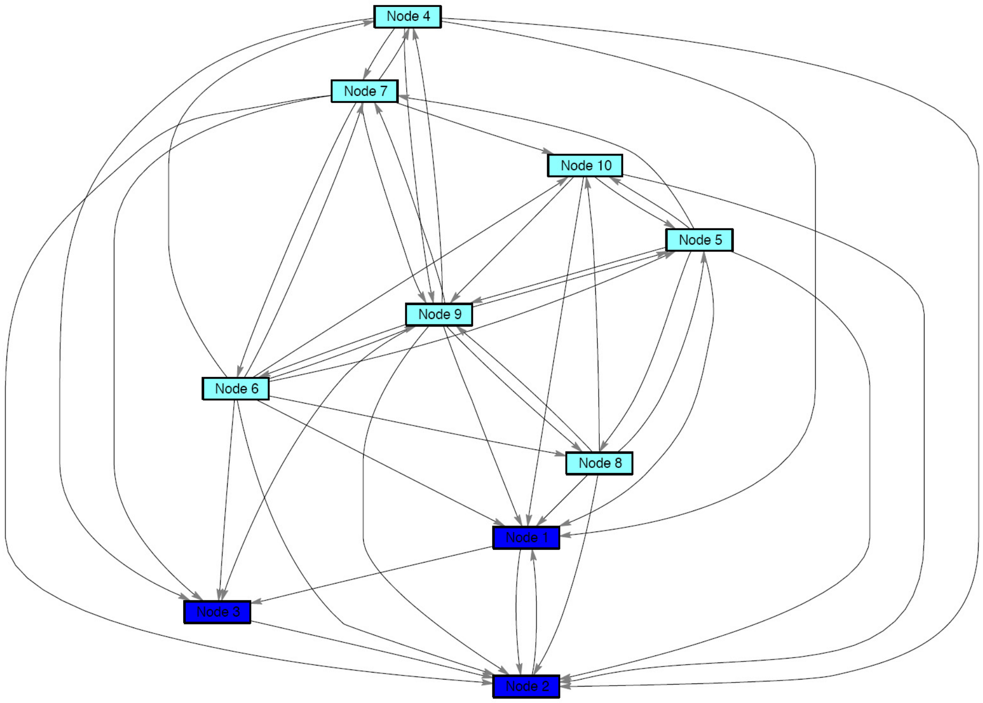

Fig. 1. The digraph $\tilde{\mathcal{G}}(\sigma)$ corresponding to the system used in the numerical example.

[5] K. L. Kosmatopoulos, E. B. Ioannou, and P. A. Ryaciotaki-Boussalis, "Large segmented telescopes: centralized decentralized and overlapping control designs," *IEEE Control Systems Magazine*, vol. 20, no. 5, pp. 59-72, 2000.

[6] D. D. Siljak, Decentralized control of complex systems, Cambridge: Academic Press, 1991.

[7] S. H. Wang and E. J. Davison, "On the stabilization of decentralized control systems," *IEEE Transactions on Automatic Control*, vol. 18,

no. 5, pp. 473-478, 1973.

[8] B. L. O. Anderson and D. J. Clements, “Algebraic characterizations of fixed modes in decentralized systems,” *Automatica*, vol. 17, no. 5, pp. 703-712, 1981.

[9] B. L. O. Anderson, “Transfer function matrix description of decentralized fixed modes,” *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1176-1182, 1982.

[10] E. J. Davison and T. N. Chang, “Decentralized stabilization and pole assignment for general proper systems,” *IEEE Transactions on Automatic Control*, vol. 35, no. 6, pp. 652-664, 1990.

[11] J. Lavaei and A. G. Aghdam, “A graph theoretic method to find decentralized fixed modes of LTI systems,” *Automatica*, vol. 43, no. 12, pp. 2129-2133, 2007.

[12] E. J. Davison and S. H. Wang, “A characterization of decentralized fixed modes in terms of transmission zeros,” *IEEE Transactions on Automatic Control*, vol. 30, no. 1, pp. 81-82, 1985.

[13] E. J. Davison, “Decentralized stabilization and regulation in large multivariable systems,” in *Directions in Large Scale Systems*, Y. C. Ho and S. K. Mitter, Eds. New York: Plenum, pp. 303-323, 1976.

[14] A. G. Aghdam and E. J. Davison, “Discrete-time control of continuous systems with approximate decentralized fixed modes,” *Automatica*, vol. 44, no. 1, pp. 75-87, 2008.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, “Introduction to Algorithms,” The MIT Press, 2nd edition, 2001.

[16] R. Tarjan, “Depth-first search and linear graph algorithms,” *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146-160, 1972.

[17] J. Cheriyan and K. Mehlhorn, “Algorithms for dense graphs and networks on the random access computer,” *Algorithmica*, vol. 15, no. 6, pp. 521-549, 1996.

[18] P. Bürgisser, M. Clausen and M. A. Shokrollahi, “Algebraic complexity theory,” Springer, 1997.

[19] D. Coppersmith and S. Winograd, “Matrix multiplication via arithmetic progressions,” *Journal of Symbolic Computation*, vol. 9, no. 3, pp. 251-280, 1990.