A dynamic graph characterisation of the fixed part of the controllable subspace of a linear structured system

van der Woude, Jacob; Commault, Christian; Boukhobza, Taha

DOI
10.1016/j.sysconle.2019.05.002

Publication date
2019

Document Version
Accepted author manuscript

Published in
Systems and Control Letters

A dynamic graph characterisation of the fixed part of the controllable subspace of a linear structured system

Jacob van der Woude[a], Christian Commault[b], Taha Boukhobza[c]

[a]*DIAM, EWI, Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE, Delft, the Netherlands. Email: j.w.vanderwoude@tudelft.nl*
[b]*Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France. Email: christian.commault@gipsa-lab.grenoble-inp.fr*
[c]*Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France. Email: taha.boukhobza@univ-lorraine.fr*

Abstract

In this paper we study linear structured systems described by means of system matrices of which only the zero/non-zero structure is known and where the non-zeros are supposed to have independent values. The structure of linear structured systems can be represented by means of various types of graphs, like directed graphs or dynamic graphs. Here we use both type of graphs because they enable us to formulate and study certain controllability properties in a uniform and straightforward way. In this paper we extend the results of a previous paper containing a partial characterisation of the fixed part of the controllable subspace of linear structured systems. This fixed part is defined as the part of the controllable subspace that is independent of the values to the non-zeros, and therefore can be seen as the robust part of the controllable subspace. It turns out that, by considering the generic dimension of the controllable subspace, a characterisation of the fixed part can be obtained. The latter dimension equals the size of the minimal set of nodes in the dynamic graph that separates between the set of input nodes and the set of final state nodes. Computing the supremal of such minimal separating sets, we are capable of characterising the fixed part. In the paper we indicate how this supremal minimal separating set can be obtained insightfully and efficiently using the recursive nature of the dynamic graph. Our results are illustrated by some meaningful examples.

Keywords: Controllable subspace, robust part, linear structured systems, graph theory, maximal linkings, minimal separators

1. Introduction

Linear structured system are linear systems of which only the zero/non-zero structure of the system matrices is known. The idea is that the non-zeros independently of each other can have any real value. A numerical realisation of a linear structured system is one in which the non-zeros are given a real value, whereas the zeros are always fixed to zero. If such a system is not controllable, the controllable subspace will vary according to the value of the non-zero entries. It is then of interest to characterise the states which are reachable from the origin with a suitable input, for any numerical realisation of the linear structured system. This invariant part, called the fixed part of the controllable subspace, consists of those vectors which are present in each numerical realisation of the controllable subspace. It turns out that this invariant part is spanned by unit vectors of the state space. The subspace spanned by these vectors can be seen as the robust part of the controllable subspace, i.e., the part that is insensitive to parameter variations.

In a previous paper, cf. [2], we formally defined and characterised this fixed part of the controllable subspace of a linear structured system. It follows easily, that the ith unit vector is in this fixed part if and only if connecting an additional input to the ith state component does not enlarge the controllability of the system, or, equivalently, does not increase the dimension of the controllable subspace. In terms of the associated graph, it is then said that the ith node of the graph is a fixed one. Therefore, a characterisation of fixed nodes in terms of properties of the underlying graph was derived. In fact, first the conditions for structural controllability are recalled from literature, as well as a characterisation for the generic dimension of the controllable subspace. In contrast to the controllability subspace itself, the dimension of the space generically does not depend on the values of the non-zero entries. In fact, this dimension can be obtained from the graph of a linear structured system.

In the literature various equivalent conditions for structural controllability can be found. In [2], we used a condition consisting of two requirements, one on connectivity in the underlying graph, and one on the rank of a certain matrix. The connectivity can be best analysed by looking at the directed graph of the underlying system, whereas the rank condition can be best analysed by looking at decompositions of the bipartite graph corresponding to the underlying system. Both requirements separately can be worked out completely, each with respect to their own type of graph (directed or bipartite).

However, the combination of the two requirements, necessitating the combination of the two types of graph, is not always possible/easy. Therefore, not a full characterisation of the fixed nodes of a linear structured system could be given in [2]. For this reason, in the current paper a characterisation of the structural controllability is used that consists of one condition only and requires just one type of graph. This makes the approach of characterising the fixed modes more natural and more easy. The drawback of the current approach is that a type of graph is required that is 'larger' than the two types of graphs used in [2]. Indeed, the type of graph that plays an important role in this paper is the dynamic graph, see Murota [8], which actually not only represents the structure of the system, but also the associated evolution in time. Because of the latter the dynamic graph consists of $n(n + m)$ nodes and nk edges, where n and m denote the number of states and inputs, respectively, and k denotes the number of non-zeros in the system matrices. The representation by the directed graph requires $n + m$ nodes and k edges, while the bipartite graph needs $2n + m$ nodes and k edges.

It turns out that structural controllability of the original linear structured system can be related with the existence of collections of disjoint paths. More precisely, the generic dimension of the controllable subspace equals the maximal number of disjoint paths between the set of input nodes and the set of final states, see Poljak [10]. Knowing the maximal number of such paths is the same as knowing the minimal number of nodes that separate the set of input nodes from the set of final state nodes. Therefore, in this paper we very much focus on such separating node sets, and, in particular, on the set that is as close as possible to the set of final state nodes. This set is referred to as the supremal minimal separator between set of input nodes and the set of final state nodes.

One of our main results is, once knowing the supremal minimal separating subset, how the set of fixed nodes can be found. The computation of the supremal minimal separator may be computationally demanding. However, using the repetitive nature of the dynamic graph, we have been able to subdivide the computations into smaller parts that give more insight and that may be more tractable from a computation point of view.

The results in [6] are used to determine the generic dimension of the controllable subspace of a structured system. However, this dimension does not say anything about the controllability of the individual nodes, as will be shown in the examples. In the current paper we use and extend the ideas and results of [6] in order to be able to precisely point out which nodes correspond to unit vectors that are always present in the controllable subspace. In this sense such nodes can be considered as structurally controllable nodes.

The outline of this paper is as follows. In Section 2 we set the scene of this paper and give the problem formulation. Section 3 contains an alternative characterisation of the fixed nodes for the type of structured systems in this paper. In Section 4 we present more information of minimal separators and some of their properties. In Section 5 we describe the algorithm to determine which nodes are fixed and which not. Section 6 contains some examples illustrating our characterisation, especially the example that we could not treat in [2]. We end the paper with a Section 7 containing conclusions and topics for future research.

2. Problem formulation

2.1. Stuctured system

We consider the discrete-time version[1] of the structured system

$$x(k+1) = Ax(k) + Bu(k), \tag{1}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^m$ the input vector, and A and B structured matrices of suitable dimensions. Here structured matrices are matrices of which only the zero/non-zero structure is known, and in which the non-zeros independently of each other may have any real value, and in which zeros are always fixed to zero. Having k non-zeros in A and B, and parametrising the i-th non-zero by a parameter $\lambda_i \in \mathbb{R}, i = 1, 2, \ldots, k$, the collection of all possible system matrices A and B can be parametrised by a vector $\lambda \in \mathbb{R}^k$. For each parameter value $\lambda \in \mathbb{R}^k$, the corresponding numerically specified system matrices are denoted by A_λ and B_λ, respectively. Hence, for every $\lambda \in \mathbb{R}^k$, the controllability matrix $C_\lambda := [B_\lambda, A_\lambda B_\lambda, \ldots, A_\lambda^{n-1} B_\lambda]$ can be determined and its rank can be computed. It turns out, cf. [10], that the rank of C_λ will have the same value for almost all $\lambda \in \mathbb{R}^k$. Here *for almost all* $\lambda \in \mathbb{R}^k$ means for all $\lambda \in \mathbb{R}^k$, except for those in some proper algebraic set

[1]The results of this paper also hold for continuous-time systems as the controllability for such systems can be expressed in the same way as for discrete-time systems.

in the parameter space $\mathbb{R}^k$. Recall that a proper algebraic subset in $\mathbb{R}^k$ is a subset of zero measure in $\mathbb{R}^k$. For more on structured systems see [3, 7, 8].

Given $\lambda \in \mathbb{R}^k$, it is also possible to determine the controllable subspace given by $\operatorname{Im} C_\lambda$. It turns out that this subspace may contain vectors that are generically independent of $\lambda \in \mathbb{R}^k$. The set of these vectors in fact forms a linear subspace in $\mathbb{R}^n$. In [2], we showed that this subspace is spanned by a number of unit vectors e_i, where e_i is the vector with a 1 at the i-th position and zeros elsewhere. This linear subspace is called the fixed part of the controllable subspace. It can be seen as the robust part of the controllable subspace, i.e., the part that is independent of the system parameters. The other part of the controllable subspace does vary with the parameter variations. It is of course useful to know which unit vectors span the fixed part of the controllable subspace as the states in that part can be controlled structurally.

2.2. Directed graphs

In this paper, as in our previous paper [2], we follow a graph theory approach to structured systems and we want to determine the unit vectors that span the fixed part of the controllable subspace by means of graph theoretic methods. For this reason, we introduce now the graph that naturally corresponds to the structured system (1). The associated *directed graph* is denoted by $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the *set of nodes* $\mathcal{X} \cup \mathcal{U}$, with the set of *state nodes* $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ and the set of *input nodes* $\mathcal{U} = \{u_1, u_2, \cdots, u_m\}$. Further, $\mathcal{E}$ is the *set of edges* given by $\{(x_j, x_i) | A_{ij} \neq 0\} \cup \{(u_j, x_i) | B_{ij} \neq 0\}$, where (x_j, x_i) denotes an edge from node x_j to node x_j, and $A_{ij} \neq 0$ indicates that the (i, j) entry of matrix A is a non-zero. Similarly, for (u_j, x_i) and $B_{ij} \neq 0$.

In G, a *path* from v_{i_0} to v_{i_q} is a sequence of edges $(v_{i_0}, v_{i_1}), (v_{i_1}, v_{i_2}), \ldots,$ $(v_{i_{q-1}}, v_{i_q}) \in \mathcal{E}$, such that $v_{i_0}, v_{i_1}, \ldots, v_{i_q} \in \mathcal{V}$, where the nodes v_{i_0} and v_{i_q} are occasionally referred to as the begin node and end node, respectively.

A path is called *simple* if any node on it occurs only once. A *cycle* is a simple path of which the begin node and end node coincide, i.e., $v_{i_0} = v_{i_q}$. A *stem* is a simple path that has its begin node in the input nodes set $\mathcal{U}$. The graph G is called *input connected* if every state node is the end node of some stem, i.e., of a path that begins in $\mathcal{U}$. Finally, we say that a *collection* of paths and/or cycles are *disjoint*, if they mutually do not have any node in common.

2.3. Structural controllability

The notion of structural controllability was introduced by Lin in [7]. There it was characterised by means of cacti spanning all state nodes, where cacti are a number of disjoint cactuses, with cactus being a special kind of subgraph, recursively made up from a stem by adding so-called buds. Structural controllability can be interpreted as generic controllability, i.e., given a parametrisation as in Subsection 2.1, it equals the controllability of the pair (A_λ, B_λ) for almost all $\lambda \in \mathbb{R}^k$.

Alternative conditions for structural controllability have been developed. For instance, see [11], system (1) is structurally controllable if and only if graph G is input connected and the set of state nodes is covered by disjoint sets of stems starting in $\mathcal{U}$ and cycles in $\mathcal{X}$. In case a system is not structurally controllable, but its graph G is input connected, the generic dimension of the corresponding controllable subspace can be determined using graph theory. Indeed, under the assumption of input connectedness, the generic dimension of the controllable subspace equals the maximal number of state nodes which can be covered by disjoint sets of stems starting in $\mathcal{U}$ and cycles in $\mathcal{X}$, cf. [6]

2.4. Fixed nodes

Throughout this document, we assume that the directed graph G, introduced in 2.2, is input connected. As indicated, our goal is to identify the unit vectors that generically span the fixed part of the controllable subspace. These unit vectors can be identified with nodes in graph G. From [2], it turns out that unit vector e_i is in the fixed part of the controllable subspace if and only if state node x_i in G is a fixed node. Here we define the state node x_i to be a fixed node[2], if, when adding a new input node $\bar{u}$ and an edge $(\bar{u}, x_i)$ to the graph G, this does not increase the maximal number of state nodes that can be covered by disjoint sets of stems starting in $\mathcal{U} \cup \{\bar{u}\}$ and cycles in $\mathcal{X}$, see [2] for more details on this. In other words, a node is fixed if the above node and edge addition does not increase the generic dimension of the controllable subspace related to (1), cf. [6].

Notice that the previous input connection assumption does not induce a real loss of generality, since it is clear that adding a new input to a non connected node will increase the size of the controllable subspace. Hence,

[2]This is shorthand for saying that unit vector e_i is in the fixed controllable subspace.

non connected nodes are non fixed nodes, and therefore we can restrict our study to the input connected part of the graph.

2.5. Unified approach

In [2], we used conditions for structural controllability that consisted of two parts, namely input connectedness and the generic rank of $[A, B]$. Both conditions individually can be analysed completely using graph methods. However, input connectedness can be best treated in the context of directed graphs, whereas the rank condition can be best treated by means of bipartite graphs. As seen in [2], the combination of the two types of graphs is not always straightforward and may lead to cases in which no results on fixed nodes could be obtained. Therefore, to avoid having two conditions, and moreover two different types of graphs, we study in this paper an alternative condition that expresses structural controllability using just one type of graph. To do so, in the remainder we will use the notion of dynamic graph, see [8] and [10].

2.6. Dynamic graph

We now introduce the *dynamic graph* associated to system (1), see [8] for more details. To define the graph G^d precisely, we introduce the node sets

$$\begin{aligned} \mathcal{X}_k &= \{x_{1k}, x_{2k}, \cdots, x_{nk}\},\ 0 < k \leqslant n, \\ \mathcal{U}_k &= \{u_{1k}, u_{2k}, \cdots, u_{mk}\},\ 0 \leqslant k < n, \end{aligned}$$

where the first index of each node stands for the component within the corresponding vector, and the second index can be interpreted as the (discrete) time. The dynamic graph can then be defined as $G^d = (\mathcal{V}^d, \mathcal{E}^d)$ with node set

$$\mathcal{V}^d = \mathcal{X}^d \cup \mathcal{U}^d, \text{ where } \mathcal{X}^d = \bigcup_{0<k\leqslant n} \mathcal{X}_k, \quad \mathcal{U}^d = \bigcup_{0\leqslant k<n} \mathcal{U}_k,$$

and edge set

$$\mathcal{E}^d = \bigcup_{1<k\leqslant n} \{(x_{j\,k-1}, x_{ik}) | A_{ij} \neq 0\} \cup \bigcup_{1\leqslant k\leqslant n} \{(u_{j\,k-1}, x_{ik}) | B_{ij} \neq 0\}.$$

Note that the set $\mathcal{X}^d$ can be restricted to a finite number of sets $\mathcal{X}_k, 0 < k \leqslant n$, for the same reason as the controllability matrix can be restricted to a finite number of matrices of the type $A^k B, 0 \leqslant k < n$, namely the finite dimensional nature of the systems under consideration. This is irrespective of whether the systems are continuous-time or discrete-time, as the controllability criterion for both type of systems is the same in terms of A and B.

2.7. Generic dimension

The dynamic graph G^d has an input nodes set $\mathcal{U}^d$ and the set $\mathcal{X}_n$ can be seen as the output nodes set. In the dynamic graph, we use the same notion of path as in graph G. Especially, we consider paths that start in $\mathcal{U}^d$ and end in $\mathcal{X}_n$. In fact, we consider collections of such paths, also called linkings, that mutually do not have a node in common.

In particular, we are interested in linkings of maximum size, i.e., the maximal number of paths from $\mathcal{U}^d$ to $\mathcal{X}_n$ that are mutually disjoint. The simple relation between this maximal number and the generic dimension of the controllable subspace was established in [10].

Proposition 1. *The generic dimension of the controllable subspace of the original system (1) equals the size of a maximal linking from $\mathcal{U}^d$ to $\mathcal{X}_n$ in the dynamic graph G^d.*

2.8. Minimal separators

Also we will consider sets that separate between $\mathcal{U}^d$ and $\mathcal{X}_n$, meaning that every path from $\mathcal{U}^d$ to $\mathcal{X}_n$ must contain at least one node in such set. According to Menger's theorem, see [8], the maximal size of a linking from $\mathcal{U}^d$ to $\mathcal{X}_n$ is equal to the minimal size of a separator between $\mathcal{U}^d$ and $\mathcal{X}_n$. Hence, by Menger's theorem (cf [8]), the generic dimension of the controllable subspace is also equal to the minimal size of a separator between $\mathcal{U}^d$ and $\mathcal{X}_n$.

Note that linkings of maximal size are not necessarily unique. The same applies to separators of minimal size. In fact, the minimal separators between $\mathcal{U}^d$ and $\mathcal{X}_n$, when possible, can be compared according to the following ordering. Let $\mathcal{S}$ and $\mathcal{T}$ be two minimal separators between $\mathcal{U}^d$ and $\mathcal{X}_n$. Then $\mathcal{T}$ is said to succeed $\mathcal{S}$ (or $\mathcal{S}$ is said to precede $\mathcal{T}$), denoted $\mathcal{S} \sqsubseteq \mathcal{T}$, if every path from $\mathcal{U}^d$ to $\mathcal{X}_n$ first passes through $\mathcal{S}$ and next passes through $\mathcal{T}$. Using this ordering, it can be shown that there always is a unique supremal (or largest) minimal separator between $\mathcal{U}^d$ and $\mathcal{X}_n$, cf. [4].

Notice that the structure of the set of separating sets has been used before in a fault detection context in [1].

3. Main results

3.1. Supremal minimal separator

Consider the *supremal minimal separator* between $\mathcal{U}^d$ to $\mathcal{X}_n$, in the remainder of this document denoted by $\mathcal{S}^*$. Every path from $\mathcal{U}^d$ to $\mathcal{X}_n$ has to pass through $\mathcal{S}^*$.

The number of nodes in $\mathcal{S}^*$ equals the size of a maximal linking from $\mathcal{U}^d$ to $\mathcal{X}_n$. In the sequel, we denote the number of nodes in $\mathcal{S}^*$ by μ. It is then easy to see that there is a linking of size μ from $\mathcal{U}^d$ to $\mathcal{S}^*$, and there is a linking of size μ from $\mathcal{S}^*$ to $\mathcal{X}_n$, where the paths in each of the linkings may have length 0. The latter can happen when $\mathcal{S}^* \cap \mathcal{X}_n \neq \varnothing$ or $\mathcal{S}^* \cap \mathcal{U}_d \neq \varnothing$. More details on this will follow in Section 4.

Supremal in the previous means that the minimal separator $\mathcal{S}^*$ between $\mathcal{U}^d$ to $\mathcal{X}_n$ is as close as possible to $\mathcal{X}_n$, i.e., there is no other minimal separator, say $T^* \neq \mathcal{S}^*$, such that some path from $\mathcal{U}^d$ to $\mathcal{X}_n$ first passes through $\mathcal{S}^*$ and next passes through T^*. Note that $\mathcal{S}^*$ is *uniquely* determined. This is due to the *lattice structure* that the set of minimal separators between $\mathcal{U}^d$ and $\mathcal{X}_n$ obeys. See also the remarks on the ordering of minimal separators in Section 2.8. More information can further be found in [12].

3.2. Preliminary result

Now consider a node x_{ik}, $0 < k \leqslant n$, on a path from $\mathcal{S}^*$ to $\mathcal{X}_n$, i.e., node x_{ik} is located in between $\mathcal{S}^*$ and $\mathcal{X}_n$. Assume, in addition, that $x_{ik} \notin \mathcal{S}^*$. Seen on a path from $\mathcal{U}^d$ to $\mathcal{X}_n$, node x_{ik} is truly after $\mathcal{S}^*$. The next lemma is instrumental in the proof of one of our main results. Recall that $\mathcal{S}^*$ consists of μ nodes.

Lemma 1. *There exists a linking of size* $\mu + 1$ *from* $\mathcal{S}^* \cup \{x_{ik}\}$ *to* $\mathcal{X}_n$.

Proof Assume on the contrary, that there is no size $\mu + 1$ linking from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$. Then the maximal size of a linking from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$ is μ. Indeed, from Section 3.1 it is clear that there a size μ linking from $\mathcal{S}^*$ to $\mathcal{X}_n$, which induces a linking of size μ from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$. To see this, note that the linking from $\mathcal{S}^*$ to $\mathcal{X}_n$ either does or does not contain x_{ik} on one of its disjoint paths. If it does not contain x_{ik}, the linking can be kept as it is. If it does contain x_{ik}, the linking can be modified by ignoring the subpath from $\mathcal{S}^*$ to x_{ik}. In both cases a linking of size μ from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$ is obtained.

Hence, our initial assumption implies that there is a size μ separator, say T^*, between $\mathcal{S}^* \cup \{x_{ik}\}$ and $\mathcal{X}_n$. Being a separator, it means that every path from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$ has to pass through T^*. Recall that every path from $\mathcal{U}^d$ to $\mathcal{X}_n$ has to pass through $\mathcal{S}^*$, and consequently also has to pass through T^*. Hence, like $\mathcal{S}^*$, also T^* is a minimal size separator between $\mathcal{U}^d$ and $\mathcal{X}_n$. However, T^* is closer to $\mathcal{X}_n$ than $\mathcal{S}^*$, i.e., $\mathcal{S}^* \sqsubseteq \mathcal{T}^*$, implying that

$\mathcal{S}^*$ can not be a *supremal* minimal separator between $\mathcal{U}^d$ and $\mathcal{X}_n$. This yields a contradiction with the basic properties of $\mathcal{S}^*$. So our initial assumption is wrong, and there does exist a linking of size $\mu + 1$ from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$. So, the proof of the claim is completed. □

Hence, for the node x_{ik} as introduced in the proof of Lemma 1, there is a size $\mu+1$ linking from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$. Consequently, in the dynamic graph G^d, after connecting additional input nodes $\bar{u}_{l-1}$ to the nodes x_{il} by edges $(\bar{u}_{l-1}, x_{il})$, for $0 < l \leqslant n$, it follows that there is a linking of (at least) size $\mu + 1$ from $\mathcal{U}^d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$, where $\bar{\mathcal{U}} = \{\bar{u}_l, 0 \leqslant l < n\}$. Indeed, take the latter mentioned size $\mu + 1$ linking from $\mathcal{S}^* \cup \{x_{ik}\}$ to $\mathcal{X}_n$, and extend/concatenate it with a linking of size μ from $\mathcal{U}^d$ to $\mathcal{S}^*$. Then a linking of size $\mu + 1$ from $\mathcal{U}^d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$ is obtained.

3.3. Main result

In the original graph G, the previous means that x_i is not a fixed node, since by connecting an additional input node $\bar{u}$ to it, through an edge $(\bar{u}, x_i)$, the dimension of the associated controllable subspace is increased, as follows from the size of maximal linking in the extended dynamic graph.

So, the conclusion is that node x_i in the original graph G is not a fixed node if there exists a $0 < k \leqslant n$ such that in the dynamic graph G^d node x_{ik} is contained on a path from $\mathcal{S}^*$ to $\mathcal{X}_n$ and $x_{ik} \notin \mathcal{S}^*$. In fact, we have the next theorem, which is one of the main results of this paper.

Theorem 1. *Node x_i in graph G is not a fixed node if and only if in G^d there exists a k, $0 < k \leqslant n$ such that node x_{ik} is contained on a path from $\mathcal{S}^*$ to $\mathcal{X}_n$ and $x_{ik} \notin \mathcal{S}^*$.*

Proof The if-part is given above. For the only if-part, note that in the extended dynamic graph $(\mathcal{V}^d \cup \bar{\mathcal{U}}, \mathcal{E}^d \cup \bar{\mathcal{E}})$, with $\bar{\mathcal{U}} = \{\bar{u}_l, 0 \leqslant l < n\}$ and $\bar{\mathcal{E}} = \{(\bar{u}_{l-1}, x_{il}), 0 < l \leqslant n\}$, there is a linking of $\mu + 1$ only if there is a node x_{ik} that is located after $\mathcal{S}^*$, seen from the perspective of paths from $\mathcal{U}^d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$. Indeed, if such node does not exist, i.e., if all nodes x_{ik} on paths from $\mathcal{U}_d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$ are before or in $\mathcal{S}^*$, then maximum size of a linking from $\mathcal{U}^d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$ is restricted by $\mathcal{S}^*$, and consequently is μ. This completes the only if-part. □

The following conjecture now follows naturally out of Theorem 1.

Conjecture 1. *Node x_i in graph G is a fixed node if and only if node x_{in} of G^d is contained in $\mathcal{S}^*$.*

The necessity of the condition in Conjecture 1 is clear. Indeed, if $x_{in} \notin \mathcal{S}^*$, there must exists a maximum linking with no path ending in x_{in}, otherwise x_{in} would belong to a separator and $\mathcal{S}^*$ would not be the supremal minimal separator. Therefore, connecting an additional input $\bar{u}_{n-1}$ to x_{in}, in G^d does increase the size of a maximal linking from $U^d \cup \bar{\mathcal{U}}$ to $\mathcal{X}_n$. The latter is equivalent to the fact that connecting an additional input $\bar{u}$ to x_i in G does increase the dimension of the controllable subspace. Hence, it follows that x_i is not fixed, and the necessity is proved. However, a proof of the sufficiency part of the condition is not clear yet, and is still subject of research.

The significance of the conjecture is that it makes the characterisation of the fixed states easier as only the states in the final time state set $\mathcal{X}_n$ need to be examined.

4. Intermezzo on general supremal minimal separators

This section contains an intermezzo on supremal minimal separators in general directed graphs. Therefore, we consider a directed graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V}$ the node set and $\mathcal{E}$ the edge set. Let paths, collections of disjoint paths, and so on, be defined as before.

4.1. Maximum size linkings

Let $\mathcal{A}$ and $\mathcal{B}$ be subsets of node set $\mathcal{V}$, and consider linkings from $\mathcal{A}$ to $\mathcal{B}$, i.e. collections of disjoint paths from $\mathcal{A}$ to $\mathcal{B}$. Note that the node sets $\mathcal{A}$ and $\mathcal{B}$ do not have to be disjoint, implying that one or more of the paths in such a linking may have length 0.

To define the maximal number of disjoint paths from $\mathcal{A}$ to $\mathcal{B}$ more clearly, we consider a suitable extension of the graph G. Indeed, we add two nodes a and b to the graph, together with edges from node a to all nodes in $\mathcal{A}$, and edges from all nodes in $\mathcal{B}$ to node b.

Then the maximal number of disjoint paths from $\mathcal{A}$ to $\mathcal{B}$ in graph G can be defined as the maximal number of paths from a to b in the extended graph that do not have a node in $\mathcal{V}$ in common with each other. Of course, all such paths share the nodes a and b. The maximal number of such disjoint paths is also referred to as the maximal size of a linking from $\mathcal{A}$ to $\mathcal{B}$.

4.2. Minimum size separators

Let $\mathcal{C}$ be an additional subset of node set $\mathcal{V}$ such that every path from node a to node b in the extended graph (or every path from $\mathcal{A}$ to $\mathcal{B}$ in G) has to pass through $\mathcal{C}$. In other words, $\mathcal{C}$ is a separator between $\mathcal{A}$ and $\mathcal{B}$. Note that also now $\mathcal{C}$ does not have to be disjoint from $\mathcal{A}$ and/or $\mathcal{B}$.

The minimal size of a separating subset between $\mathcal{A}$ and $\mathcal{B}$ is equal to the maximal size of a linking from $\mathcal{A}$ to $\mathcal{B}$. This is due to the Menger's theorem, cf. [8].

Let $\mathcal{S}$ be a minimal size separator between $\mathcal{A}$ and $\mathcal{B}$, then $|\mathcal{S}| \leqslant |\mathcal{A}|, |\mathcal{S}| \leqslant |\mathcal{B}|$, where $|\mathcal{S}|$ denotes the number of elements of $\mathcal{S}$, and similarly for $|\mathcal{A}|$ and $|\mathcal{B}|$. Further, there exists a size $|\mathcal{S}|$ linking from $\mathcal{A}$ to $\mathcal{S}$, and a size $|\mathcal{S}|$ linking from $\mathcal{S}$ to $\mathcal{B}$. Note that in these linkings there may be paths of length 0.

4.3. Partial ordering

Considering all minimal size separators between $\mathcal{A}$ and $\mathcal{B}$, there exists a supremal (and an infimal) minimal separator between $\mathcal{A}$ and $\mathcal{B}$, because the set of these minimal separators has a lattice structure. This follows from the work of Escalant [4].

The ordering underlying the lattice structure is as follows. Given two minimal separators $\mathcal{S}$ and $\mathcal{T}$ between $\mathcal{A}$ and $\mathcal{B}$, then $\mathcal{T}$ is said to succeed $\mathcal{S}$ (or $\mathcal{S}$ precedes $\mathcal{T}$), denoted $\mathcal{S} \sqsubseteq \mathcal{T}$, if every path from a to b, first passes through $\mathcal{S}$, and next passes through $\mathcal{T}$.

Note that not every two minimal separators $\mathcal{S}$ an $\mathcal{T}$ can be compared in the above way. Indeed, consider the graph G (without the nodes a and b)

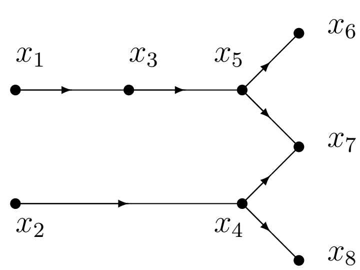

Figure 1.

with $\mathcal{A} = \{x_1, x_2\}$, $\mathcal{B} = \{x_6, x_7, x_8\}$, and $\mathcal{S} = \{x_1, x_4\}$, $\mathcal{T} = \{x_2, x_3\}$. Then $\mathcal{S}$ and $\mathcal{T}$ can not be ordered as indicated above. However, $\mathcal{P} = \{x_3, x_4\}$ and $\mathcal{Q} = \{x_4, x_5\}$ are such that $\mathcal{S} \sqsubseteq \mathcal{P}, \mathcal{T} \sqsubseteq \mathcal{P}$ and $\mathcal{S} \sqsubseteq \mathcal{Q}, \mathcal{T} \sqsubseteq \mathcal{Q}$. In fact, $\mathcal{Q}$ is the supremal minimal separator between $\mathcal{A}$ and $\mathcal{B}$.

In general, we denote the supremal minimal separator between $\mathcal{A}$ and $\mathcal{B}$ by $S^{sup}(\mathcal{A}, \mathcal{B})$.

4.4. Semi group property

Now we consider the subsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, as in the previous subsection, i.e., consider linkings from $\mathcal{A}$ to $\mathcal{B}$, and let $\mathcal{C}$ be a separator between $\mathcal{A}$ and $\mathcal{B}$. Hence, we consider collections of disjoint paths from $\mathcal{A}$ to $\mathcal{B}$, where each path from $\mathcal{A}$ to $\mathcal{B}$ has to pass through $\mathcal{C}$.

First, we focus on minimal size separators between $\mathcal{A}$ and $\mathcal{C}$, and especially on the supremal one, say $\mathcal{S}_1$, i.e., $\mathcal{S}_1 := S^{sup}(\mathcal{A}, \mathcal{C})$. Note that $\mathcal{S}_1$ separates between $\mathcal{A}$ and $\mathcal{C}$, but also between $\mathcal{A}$ and $\mathcal{B}$, because $\mathcal{C}$ is a separator between $\mathcal{A}$ and $\mathcal{B}$. Further, note that there exists a size $|\mathcal{S}_1|$ linking from $\mathcal{A}$ to $\mathcal{C}$, which can be split into a size $|\mathcal{S}_1|$ linking from $\mathcal{A}$ to $\mathcal{S}_1$, and a size $|\mathcal{S}_1|$ linking from $\mathcal{S}_1$ to $\mathcal{C}$.

Next extending the paths beyond $\mathcal{C}$ in the direction of $\mathcal{B}$, consider the maximal size linkings from $\mathcal{S}_1$ to $\mathcal{B}$, and consider the corresponding supremal minimal separator between $\mathcal{S}_1$ and $\mathcal{B}$, denoted by $\mathcal{S}_2 := S^{sup}(\mathcal{S}_1, \mathcal{B})$. Then there exists a size $|\mathcal{S}_2|$ linking from $\mathcal{S}_1$ to $\mathcal{B}$, which can be split into a size $|\mathcal{S}_2|$ linking from $\mathcal{S}_1$ to $\mathcal{S}_2$, and a size $|\mathcal{S}_2|$ linking from $\mathcal{S}_1$ to $\mathcal{B}$.

Note that the first of the last two linking implies that $|\mathcal{S}_1| \geqslant |\mathcal{S}_2|$. Moreover, note that the size $|\mathcal{S}_2|$ linking from $\mathcal{S}_1$ to $\mathcal{B}$ can be concatenated with the previously mentioned size $|\mathcal{S}_1|$ linking from $\mathcal{A}$ to $\mathcal{S}_1$, yielding a size $|\mathcal{S}_2|$ linking from $\mathcal{A}$ to $\mathcal{B}$. Note that $\mathcal{S}_2$ separates between $\mathcal{S}_1$ and $\mathcal{B}$, but also between $\mathcal{A}$ and $\mathcal{B}$. Indeed, recall that $\mathcal{S}_1$ is as separator between $\mathcal{A}$ and $\mathcal{B}$, and therefore so is $\mathcal{S}_2$.

Since, by the above linking, the maximal size of a linking between $\mathcal{A}$ and $\mathcal{B}$ is at least $|\mathcal{S}_2|$, also the minimal size of a separator between $\mathcal{A}$ and $\mathcal{B}$ is at least $|\mathcal{S}_2|$. Hence, with $\mathcal{S}_2$ being a separator between $\mathcal{A}$ and $\mathcal{B}$, it is in fact a minimal one. Moreover, $\mathcal{S}_2$ is a supremal minimal separator between $\mathcal{A}$ and $\mathcal{B}$. Indeed, if not, there exists a minimal size separator between $\mathcal{A}$ and $\mathcal{B}$, say $\mathcal{T}_2 \neq \mathcal{S}_2$, such that $\mathcal{S}_2 \sqsubseteq \mathcal{T}_2$. However, then $\mathcal{T}_2$ is also a supremal minimal separator between $\mathcal{S}_1$ and $\mathcal{B}$, with $\mathcal{S}_2 \neq \mathcal{T}_2$ implying that $\mathcal{S}_2$ is not supremal between $\mathcal{S}_1$ and $\mathcal{B}$. This yields a contradiction with the properties of $\mathcal{S}_2$. Hence, $\mathcal{S}_2$ is a supremal minimal separator between $\mathcal{A}$ and $\mathcal{B}$.

In summary, $\mathcal{S}_2 = S^{sup}(\mathcal{A}, \mathcal{B})$, where $\mathcal{S}_1 = S^{sup}(\mathcal{A}, \mathcal{C})$ and $\mathcal{S}_2 = S^{sup}(\mathcal{S}_1, \mathcal{B})$. Combining the previous, it follows that

$$\mathcal{S}^{sup}(\mathcal{A}, \mathcal{B}) = S^{sup}(S^{sup}(\mathcal{A}, \mathcal{C}), \mathcal{B}), \tag{2}$$

for all subsets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ of $\mathcal{V}$, where $\mathcal{C}$ is a separator between $\mathcal{A}$ and $\mathcal{B}$.

4.5. Computational complexity

Consider again the graph $G = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$, edge set $\mathcal{E}$, and $\mathcal{A}$ and $\mathcal{B}$ subsets of $\mathcal{V}$. Let the size of a maximal linking from $\mathcal{A}$ to $\mathcal{B}$, or, equivalently, the size of a minimal separator between $\mathcal{A}$ and $\mathcal{B}$, be given by μ. It is well-known, cf. [8, 13], that finding a maximal linking is equivalent to solving the maximum flow problem on an associated graph G_a, which is obtained from G by splitting each node x_i into two nodes x_i^+ and x_i^-, connected by an edge (x_i^+, x_i^-) of capacity one. A source node is connected to the vertices of $\mathcal{A}$, and all vertices of $\mathcal{B}$ are connected to a sink node. All the edges, except the edges resulting from the duplication of a node, have infinite capacity. When applying the well-known Ford-Fulkerson algorithm [5] to find the maximum flow in G_a, where all edges have been reversed, the first minimum cut that will be met in the augmenting path procedure corresponds exactly with the supremal minimal separator $\mathcal{S}^*$ in G. The complexity of the Ford-Fulkerson algorithm with integer capacities is $O(N_{\mathcal{E}} \cdot f_{\max})$, where $N_{\mathcal{E}}$ is the number of edges of the graph G_a, and $f_{\max}$ is the value of the maximum flow. The number of edges in G_a is bounded by $(|\mathcal{E}| + 3|\mathcal{V}|)$, and the flow being bounded by $|\mathcal{V}|$, we finally get a complexity in $O(|\mathcal{V}|^3)$, since $O(|\mathcal{E}| + 3|\mathcal{V}|) \leqslant O(|\mathcal{V}|^2)$, as $\mathcal{E}$ may consist of all edges between all nodes. One could certainly suggest some better performing maximum flow algorithms, but it is important to note that the Ford-Fulkerson algorithm also provides in one step with the supremal minimal separator $\mathcal{S}^*$. Notice that this correspondence between the maximum linking problem and the maximum flow induces relations between the analysis of the structure of the set of separators and the set of minimum cuts in max flow problems, cf. [9].

5. Algorithm to compute $\mathcal{S}^*$

In this section we return to the directed graph G and dynamic graph G^d, corresponding to structured system (1). For the directed graph G, we have that $|\mathcal{V}| = O(n)$ and $|\mathcal{E}| = O(n^2)$, assuming that in the structured system (1) the number of states n is (much) larger than the number of inputs m, and that both A and B may be full matrices. For the dynamic graph G^d, this implies that $|\mathcal{V}^d| = O(n^2)$ and $|\mathcal{E}^d| = O(n^3)$. Then by Section 4.5, $\mathcal{S}^*$ can be obtained by a computation of order $O(n^5)$. Note that in case the system matrices A and B are sparse matrices, it is often true that $|\mathcal{E}| = O(n)$ and $|\mathcal{E}^d| = O(n^2)$, so that $\mathcal{S}^*$ can be obtained by a computation of order $O(n^4)$.

Instead of computing $\mathcal{S}^*$ directly, it may be profitable to use the repetitive structure of the dynamic graph, breaking up the computation into smaller parts, that each can be done with smaller number of operations, so that the overall computation can be done with less computations. This breaking up will be explained in more detail in the next subsections.

5.1. Extra notation

Given a node set S in the dynamic graph, denote $\sigma^{-1}S = \{x_{i\,j} | x_{i\,j+1} \in S\}$. Hence, $\sigma^{-1}S$ is a version of S that is shifted one to the left (backwards in time). Note that in general it is possible that $S \cap \sigma^{-1}S \neq \emptyset$. Examples are $\sigma^{-1}\mathcal{X}_n = \mathcal{X}_{n-1}$, $\sigma^{-1}\mathcal{U}_{n-1} = \mathcal{U}_{n-2}$, and in case $S = \{x_{1,4}, x_{1,3}, x_{4,4}, x_{5,4}, x_{6,3}\}$, then $\sigma^{-1}S = \{x_{1,3}, x_{1,2}, x_{4,3}, x_{5,3}, x_{6,2}\}$.

5.2. Preliminary observations

With the previous notation, the goal is to compute $\mathcal{S}^* = S^{sup}(\mathcal{U}^d, \mathcal{X}_n)$, where $\mathcal{U}^d = \mathcal{U}_0 \cup \mathcal{U}_1 \cup \cdots \cup \mathcal{U}_{n-1}$.

Define $S_k := S^{sup}(\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$, for $0 \leqslant k < n$. In words, S_k is the supremal minimal separator between the disjoint node sets $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-1}$ and $\mathcal{X}_n$. Note that $\mathcal{S}^* = \mathcal{S}_0$.

From the above definition, it follows that $S_{n-1} = S^{sup}(\mathcal{U}_{n-1}, \mathcal{X}_n)$ and $S_{k+1} = S^{sup}(\mathcal{U}_{k+1} \cup \mathcal{U}_{k+2} \cup \cdots \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$. Note that S_{n-1} can be simply obtained by a straightforward, DM-like, decomposition, cf. [8].

It follows by invariance, by shifting everything one step backwards in time, that $\sigma^{-1}S_{k+1} = S^{sup}(\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2}, \mathcal{X}_{n-1})$. Hence, $\sigma^{-1}S_{k+1}$ is the supremal minimal separator between the disjoint node sets $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2}$ and $\mathcal{X}_{n-1}$.

Note that $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2}$ and $\mathcal{X}_{n-1}$ are both disjoint from $\mathcal{U}_{n-1}$. Also no edges begin in $\mathcal{U}_{n-1}$, and end in $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2}$ or in $\mathcal{X}_{n-1}$, and vice versa.

Hence, adding $\mathcal{U}_{n-1}$ to both the begin node set and the end node set, it follows that $\sigma^{-1}S_{k+1} \cup \mathcal{U}_{n-1}$ is the supremal minimal separator between $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}$ and $\mathcal{X}_{n-1} \cup \mathcal{U}_{n-1}$. So, with the latter it follows that $\sigma^{-1}S_{k+1} \cup \mathcal{U}_{n-1} = S^{sup}(\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}, \mathcal{X}_{n-1} \cup \mathcal{U}_{n-1})$.

Now define the node set $T_k := S^{sup}(\sigma^{-1}S_{k+1} \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$ with, as above, $S_{k+1} = S^{sup}(\mathcal{U}_{k+1} \cup \mathcal{U}_{k+2} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$. After substitution, it follows that $T_k = S^{sup}\big(S^{sup}(\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}, \mathcal{X}_{n-1} \cup \mathcal{U}_{n-1}), \mathcal{X}_n\big)$. Note that $\mathcal{X}_{n-1} \cup \mathcal{U}_{n-1}$ is a separator between $\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}$ and $\mathcal{X}_n$.

Then using (2) with $\mathcal{A} = \mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}$, $\mathcal{B} = \mathcal{X}_n$ and $\mathcal{C} = \mathcal{X}_{n-1} \cup \mathcal{U}_{n-1}$, it follows that $T_k = S^{sup}(\mathcal{U}_k \cup \mathcal{U}_{k+1} \cup \cdots \cup \mathcal{U}_{n-2} \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$. Hence, $S_k = T_k$, and consequently $S_k = S^{sup}(\sigma^{-1} S_{k+1} \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$, for $0 \leqslant k < n-1$.

5.3. Algorithm to compute $\mathcal{S}^*$

The previous observations lead to the following algorithm.

(i) The initialization $k := n-1, S_k := S^{\text{sup}}(\mathcal{U}_{n-1}, \mathcal{X}_n)$.

(ii) Repeat

- $k := k - 1$,
- $S_k := S^{\text{sup}}(\sigma^{-1} S_{k+1} \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$.

until $k = 0$, or until $S_k = S_{k+1}$, since then convergence is achieved.

(iii) Then $\mathcal{S}^* = S_0 = S^{sup}(\mathcal{U}_0 \cup \mathcal{U}_1 \cup \cdots \cup \mathcal{U}_{n-1}, \mathcal{X}_n)$.

The initialisation and repetition in the algorithm follow from the previous observations. The completion follows in fact from the work of Poljak [10].

5.4. Computational aspects

The idea behind the breaking up of the algorithm to get $\mathcal{S}^*$, as it is done in Subsection 5.3, is to use the repetitive structure of the dynamic graph to increase the efficiency of the computation. Indeed, it is expected that the computation of the intermediate supremal minimal separators $\mathcal{S}_k$, $0 \leqslant k < n$, in practice, can be done by means of computations of order $O(n\, n^2) = O(n^3)$, because each 'layer' has $O(n)$ nodes and $O(n^2)$ edges (for full matrices). Then, repeating the computations recursively n times, it follows that the computation of $\mathcal{S}^*$ by the algorithm in Subsection 5.3 amounts to a computation of order $O(n^4)$. When the system matrices are sparse, the set $\mathcal{S}^*$ is expected to be computed by means of a computation of order $O(n^3)$, as each layer then has $O(n)$ nodes and $O(n)$ edges.

Of course, the precise computations per iteration step depend on the system under consideration. For this reason an accurate expression of the overall number of computations is not possible in general. Only a rough estimate can be given.

Nevertheless, it seems that breaking up the computation improves the efficiency as the number of computations is decreased from $O(n^5)$ to $O(n^4)$

for full system matrices, and from $O(n^4)$ to $O(n^3)$ for sparse system matrices. In addition, the breaking up into a number of steps does increase the insight in the process of computing $\mathcal{S}^*$ and location of the fixed nodes.

6. Examples

6.1. Example 1

The following example is based on the example that could not be treated by the methods in our previous paper [2]. Consider the structured system represented by the graph G, displayed in Figure 2.

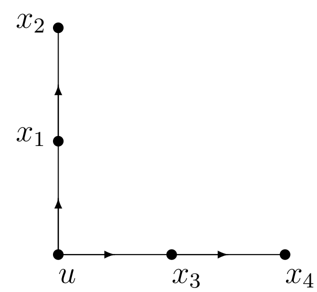

Figure 2.

The graph has input node set $\mathcal{U} = \{u\}$ and state node set $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$. The associated structured matrices A and B easily follow from the graph, as well as parametrised versions of these matrices given by A_λ and B_λ. Also the associated controllability matrix C_λ can be determined. Here, for instance,

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 \end{bmatrix}, \; B = \begin{bmatrix} * \\ 0 \\ * \\ 0 \end{bmatrix},$$

with $*$ denoting non-zero entries, and a parametrised version together with the associated controllability matrix

$$A_\lambda = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \end{bmatrix}, \; B_\lambda = \begin{bmatrix} \lambda_3 \\ 0 \\ \lambda_4 \\ 0 \end{bmatrix}, \; C_\lambda = \begin{bmatrix} \lambda_3 & 0 & 0 & 0 \\ 0 & \lambda_1\lambda_3 & 0 & 0 \\ \lambda_4 & 0 & 0 & 0 \\ 0 & \lambda_2\lambda_4 & 0 & 0 \end{bmatrix},$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$ are parametrisations of the non-zeros, and $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 \end{bmatrix}^\top \in \mathbb{R}^4$ is the overall parameter vector, with $^\top$ denoting 'transpose'. From C_λ it is clear the its rank is 2 for almost all $\lambda \in \mathbb{R}^4$. Indeed, the rank of C_λ is less than 2 for $\lambda \in \mathbb{R}^4$ such that $(\lambda_3^2 + \lambda_4^2)(\lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_4^2) = 0$, which clearly forms a proper algebraic variety in $\mathbb{R}^4$. Hence, the generic dimension of the controllable subspace is 2.

Note that the graph G is input connected. Using the main result of Hosoe [6], it then also follows that the controllable subspace had generic dimension equal to 2. Indeed, the maximum number of state nodes covered by disjoint sets of stems starting in $\mathcal{U}$ and cycles in $\mathcal{X}$ is equal to 2. Consider, for instance, the stem made up of the edges (u, x_1) and (x_1, x_2). Alternatively, the stem made up of the edges (u, x_3) and (x_3, x_4) can be considered.

Further, it can be seen quickly that when adding a control $\bar{u}$ to either node x_1 or node x_3, the generic dimension of the controllable subspace will increase to 4. Indeed, when adding a control node $\bar{u}$ to node x_1, consider the two disjoint stems $(u, x_3), (x_3, x_4)$ and $(\bar{u}, x_1), (x_1, x_2)$. Similarly, when adding a control node $\bar{u}$ to node x_3. In the same sprit, it follows that adding a control node $\bar{u}$ to either node x_2 or node x_4, the generic dimension will increase to 3. Indeed, when adding a control node $\bar{u}$ to node x_2, consider the two disjoint stems $(u, x_3), (x_3, x_4)$ and $(\bar{u}, x_2)$. Similarly, when adding a control node $\bar{u}$ to node x_4. Hence, with each of these control node additions, the generic dimension of the controllable subspace always increases. Therefore, none of state nodes is fixed.

The previous conclusion could not be obtained by the methods in [2]. The reason for this being the fact that the conditions in the latter reference require the combination of two types of graphs that not always can be combined (easily), like, for instance, in this example. Fortunately, with the method in this paper, requiring just one type of graph, it is always possible to fully investigate which nodes are fixed and which are not. Therefore, consider the dynamic graph G^d of the system corresponding to graph G, displayed in Figure 3.

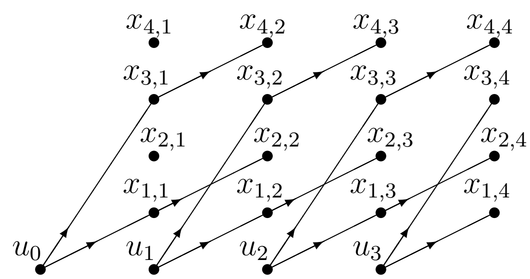

Figure 3.

Applying the algorithm for $n = 4$, it follows after the first step that $k = 3$ and $\mathcal{S}_3 = \{u_3\}$. Indeed, focussing on all paths from $\mathcal{U}_3 = \{u_3\}$ and $\mathcal{X}_4 = \{x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}\}$ in Figure 3, and ignoring all other edges and paths, it follows that the supremal minimal separator between $\mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_3 = \{u_3\}$. In the next step, $k = 2$ and $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_3 = \{u_2, u_3\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ then equals $\mathcal{S}_2 = \{u_2, u_3\}$. Next, $k = 1$ and $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_3 = \{u_1, u_2, u_3\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_1 = \{u_2, u_3\}$. Hence, $\mathcal{S}_1 = \mathcal{S}_2$, implying that $\mathcal{S}_0 = \mathcal{S}^* = \{u_2, u_3\}$. Now observe that the nodes $x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}, x_{1,3}, x_{3,3}$ are not in $\mathcal{S}^*$, and are contained in paths from $\mathcal{S}^*$ to $\mathcal{X}_4$. From Theorem 1 it then follows that the nodes x_1, x_2, x_3, x_4 in G are not fixed. Hence, none of the state nodes are fixed.

6.2. Example 2

Consider the structured system represented by the graph G, displayed in Figure 4.

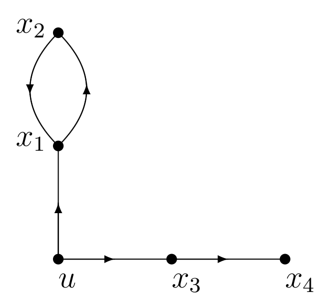

Figure 4.

The graph has input node set $\mathcal{U} = \{u\}$ and state node set $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$. The associated matrices A and B easily follow from the graph. The same applies to parametrised version of these matrices and the corresponding controllability matrix. For instance,

$$A_\lambda = \begin{bmatrix} 0 & \lambda_5 & 0 & 0 \\ \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \end{bmatrix}, \ B_\lambda = \begin{bmatrix} \lambda_3 \\ 0 \\ \lambda_4 \\ 0 \end{bmatrix}, \ C_\lambda = \begin{bmatrix} \lambda_3 & 0 & \lambda_1\lambda_3\lambda_5 & 0 \\ 0 & \lambda_1\lambda_3 & 0 & \lambda_1^2\lambda_3\lambda_5 \\ \lambda_4 & 0 & 0 & 0 \\ 0 & \lambda_2\lambda_4 & 0 & 0 \end{bmatrix},$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5 \in \mathbb{R}$ are parametrisations of the non-zeros, and $\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \end{bmatrix}^\top \in \mathbb{R}^5$ is the overall parameter vector. From C_λ it is clear the its rank is 4 for almost all $\lambda \in \mathbb{R}^5$. Hence, the generic dimension of the controllable subspace is 4, i.e., the system is generically controllable.

Clearly, the graph G is input connected. Using the main result of Hosoe [6], it follows that the controllable subspace had generic dimension equal to 4, i.e., the system is generically controllable. Indeed, the maximum number of state nodes covered by disjoint sets of stems starting in $\mathcal{U}$ and cycles in $\mathcal{X}$ is equal to 4. For this, consider the stem made up of the edges (u, x_3) and (x_3, x_4), and the cycle made up of the edges (x_1, x_2) and (x_2, x_1). Hence, adding a control to either node x_1, x_2, x_3 or x_4 will not increase the generic dimension of the controllable subspace, since it is already maximal. Therefore, all state nodes in G are fixed.

With our method it is also possible to investigate which nodes of the graph in Figure 4 are fixed and which not. Therefore, consider the dynamic graph G^d of the corresponding system, displayed in Figure 5.

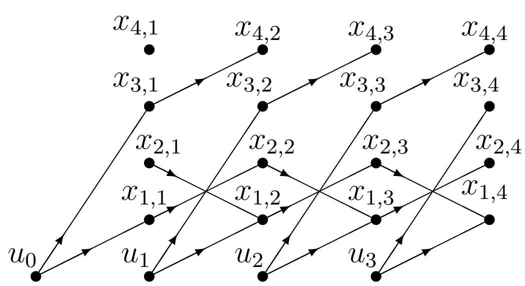

Figure 5.

Applying the algorithm for $n = 4$, it follows after the first step that $k = 3$ and $\mathcal{S}_3 = \{u_3\}$. In the next step, $k = 2$ and $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_3 = \{u_2, u_3\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_2 = \{u_2, u_3\}$. Then, $k = 1$ and $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_3 = \{u_1, u_2, u_3\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_1 = \{x_{1,4}, x_{3,4}, u_2\}$. Finally, $k = 0$ and $\sigma^{-1}\mathcal{S}_1 \cup \mathcal{U}_3 = \{x_{1,3}, x_{3,3}, u_1, u_3\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_1 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_0 = \{x_{1,4}, x_{2,4}, x_{3,4}, x_{4,4}\}$. Indeed, it is easy to see that there is a linking of size 4 from $\sigma^{-1}\mathcal{S}_1 \cup \mathcal{U}_3$ and $\mathcal{X}_4$. Actually, it is a unique one. Hence, the supremal minimal separating set coincides with $\mathcal{X}_4$. As $k = 0$, it follows that $\mathcal{S}_0 = \mathcal{S}^* = \mathcal{X}_4$. Now observe that there are no nodes outside $\mathcal{S}^*$ contained in paths from $\mathcal{S}^*$ to $\mathcal{X}_4$. From Theorem 1 it then follows that all state nodes in G are fixed.

6.3. Example 3

Consider the structured system represented by the graph G, displayed in Figure 6.

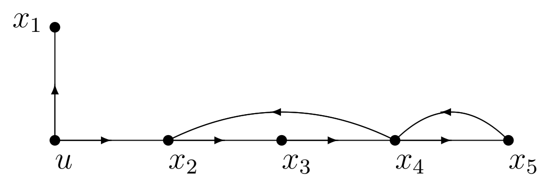

Figure 6.

The graph has input set $\mathcal{U} = \{u\}$ and state node set $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$. Clearly, the graph is input connected. Using the main result of Hosoe [6], it follows that the controllable subspace had generic dimension equal to 4. Indeed, the maximum number of state nodes covered by disjoint sets of stems starting in $\mathcal{U}$ and cycles in $\mathcal{X}$ is equal to 4. Consider, for instance, the stem made up of the edges $(u, x_2), (x_2, x_3), (x_3, x_4)$, and (x_4, x_5) , or the stem (u, x_1) combined with the cycle containing the edges $(x_2, x_3), (x_3, x_4)$ and (x_4, x_2). It can be seen easily that adding a control to either node x_3 or x_4 will not increase the generic dimension of the controllable subspace. Adding a control to either x_1, x_2 or x_5, the generic dimension of the controllable subspace does increase to 5. Hence, the nodes x_3 and x_4 are fixed, while the nodes x_1, x_2 and x_5 are not.

With our method it is possible to analyse the system for fixed and non-fixed nodes. Therefore, consider the dynamic graph G^d of the corresponding system, displayed in Figure 7.

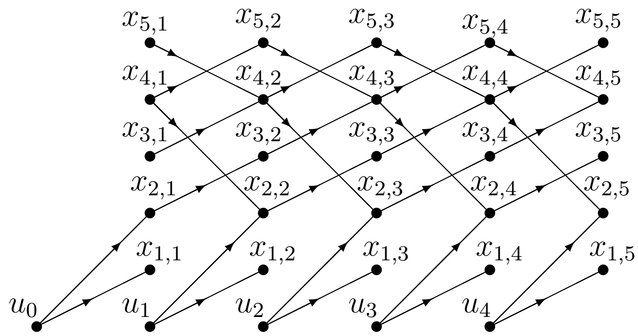

Figure 7.

Applying the algorithm for $n = 5$, it follows after the first step that $k = 4$ and $\mathcal{S}_4 = \{u_4\}$. In the next step, $k = 3$ and $\sigma^{-1}\mathcal{S}_4 \cup \mathcal{U}_4 = \{u_3, u_4\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_4 \cup \mathcal{U}_4$ and $\mathcal{X}_5$ is equal to $\mathcal{S}_3 = \{x_{3,5}, u_4\}$. Next, $k = 2$ and $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_4 = \{x_{3,4}, u_3, u_4\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_3 \cup \mathcal{U}_4$ and $\mathcal{X}_5$ is equal to $\mathcal{S}_2 = \{x_{3,5}, x_{4,5}, u_4\}$. Then, $k = 1$ and $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_4 = \{x_{3,4}, x_{4,4}, u_3, u_4\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_2 \cup \mathcal{U}_4$ and $\mathcal{X}_5$ is equal to $\mathcal{S}_1 = \{x_{4,4}, x_{3,5}, x_{4,5}, u_4\}$. Finally, $k = 0$ and $\sigma^{-1}\mathcal{S}_1 \cup \mathcal{U}_3 = \{x_{4,3}, x_{3,4}, x_{4,4}, u_3, u_4\}$. The supremal minimal separator between $\sigma^{-1}\mathcal{S}_1 \cup \mathcal{U}_3$ and $\mathcal{X}_4$ is equal to $\mathcal{S}_0 = \{x_{4,4}, x_{3,5}, x_{4,5}, u_4\}$. As $k = 0$, it follows that $\mathcal{S}_0 = \mathcal{S}^*$. The latter also follows because $\mathcal{S}_0 = \mathcal{S}_1$. Now observe that the nodes $x_{1,5}, x_{2,5}$ and $x_{5,5}$ are not in $\mathcal{S}^*$ and are contained in paths from $\mathcal{S}^*$ to $\mathcal{X}_5$. From Theorem 1 it then follows that the nodes x_1, x_2 and x_5 in G are not fixed.

6.4. Remark

Note that all three example also illustrate the potential validity of Conjecture 1. Indeed, in Example 1, $\mathcal{S}^* = \{u_2, u_3\}$ and $\mathcal{S}^* \cap \mathcal{X}_4 = \varnothing$, implying that there are no fixed nodes. In Example 2, $\mathcal{S}^* = \mathcal{X}_4$. So, $\mathcal{S}^* \cap \mathcal{X}_4 = \mathcal{X}_4$, implying that all nodes are fixed. In Example 3, $\mathcal{S}^* = \{x_{4,4}, x_{3,5}, x_{4,5}, u_4\}$. So, $\mathcal{S}^* \cap \mathcal{X}_5 = \{x_{3,5}, x_{4,5}\}$, implying that nodes x_3 and x_4 are fixed, and nodes x_1, x_2 and x_5 are not fixed.

7. Conclusions and outlook

In this paper we studied linear structured systems and focussed on the fixed part of the controllable subspace of such systems. This part consists of unit vectors that that are present in the controllable subspace, independently of the value of the non-zeros, i.e., the free parameters, in the system matrices. This part can be seen as the robust part of the controllable subspace. The index of the unit vectors can be obtained using the dynamic graph underlying the structured system. The graph theory method for achieving this, developed in this paper, amounts to finding the supremal minimal separator between the set of input nodes $\mathcal{U}^d$ and the set of final state nodes $\mathcal{X}_n$. To compute this minimal separating set a recursive algorithm has been developed making use of the repetitive nature of the dynamic graph. The algorithm breaks down the task of computing the overall supremal minimal separator into smaller pieces that each give more insight. The current paper completes the results of a previous paper by the authors, cf. [2]. By some examples, the results of the current paper are illustrated. It is expected, see also Subsection 5.4, that breaking down the computations makes that the supremal minimal separator can be obtained in a more efficient way than when starting from the complete dynamic graph. However, the precise meaning of this statement is not yet clear, and is topic of current research. A second topic of current research is Conjecture 1, stating that node x_i in graph G is not fixed if and only if node x_{in} of G^d is not contained in $\mathcal{S}^*$, or, put differently, node x_i in graph G is fixed if and only if node x_{in} of $G^d \in \mathcal{S}^*$, where $\mathcal{S}^*$ is the supremal minimal separator in G^d between $\mathcal{U}^d$ and $\mathcal{X}_n$.

8. References

[1] C. Commault, J. M. Dion, and S. Yacoub. Structural analysis for the sensor location problem in fault detection and isolation. *Automatica*, 44(8):2074–2080, 2008.

[2] C. Commault, J. van der Woude, and T. Boukhobza. On the fixed controllable subspace in linear structured systems. *Systems and Control Letters*, 102:42–47, 2017.

[3] J. M. Dion, C. Commault, and J. van der Woude. Generic properties and control of linear structured systems: a survey. *Automatica*, 39(7):1125–1144, 2003.

[4] F. Escalante. Schnittverbände in graphen. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 38(1):199–220, 1972.

[5] L.R. Ford and D.R. Fulkerson. *Flows in networks*. Princeton University Press, 1962.

[6] S. Hosoe. Determination of generic dimensions of controllable subspaces and its application. *IEEE Trans. Automat. Control*, AC-25:1192–1196, 1980.

[7] C. T. Lin. Structural controllability. *IEEE Trans. Automat. Control*, 19(3):201–208, 1974.

[8] K. Murota. *Systems Analysis by Graphs and Matroids*. Algorithms and Combinatorics. Springer-Verlag New-York, Inc., 1987.

[9] J. C. Picard and M. Queyranne. On the structure of all minimum cuts in a network and applications. *Mathematical Progamming Study*, 13:8–16, 1980.

[10] S. Poljak. On the generic dimension of controllable subspaces. *IEEE Trans. Automat. Control*, 35(3):367–369, 1990.

[11] K.J. Reinschke. *Multivariable control: a graph-theoretic approach.* Springer Verlag, 1988.

[12] J.W. van der Woude. The generic number of invariant zeros of a structured linear system. *SIAM J. Cont. Optim.*, 38(1):1–21, 2000.

[13] T. Yamada. A network flow algorithm to find an elementary I/O matching. *Networks*, 18:105–109, 1988.