Strong structural controllability of trees*

Maguy Trefois[1] and Jean-Charles Delvenne[1,2]

***Abstract*— This paper examines the strong structural controllability of networked systems. We use the notion of zero forcing set, a concept borrowed from combinatorial matrix theory, in order to solve the problem of determining a minimum-size input node set S so that a system with a self-damped/undamped tree structure is strongly S-controllable.**

I. INTRODUCTION

In many real systems, the interaction strengths between the different components of the system are unknown or only partially known. In such a case, determining the controllability of the system from classical Kalman controllability condition becomes inapplicable. An alternative is to use the structure of the system in order to get information about its controllability. That is why weak and strong structural controllability has been introduced [1], [2].

Weak structural controllability can be studied from the maximum matchings in the network [3] and provides a lower bound on the minimum number of nodes which have to be controlled for classical control over the system. Instead, the strong structural controllability (or strong controllability for short) can be determined from constraint matchings in a bipartite graph defined from the network [4] and it provides an upper bound on the minimum number of nodes to be controlled for classical control of the whole system.

Given a networked system and an outside controller on this system, the input node set is the set of nodes whose state is directly influenced by the controller. In the framework of the strong structural controllability, the tricky point is finding a minimum-size node set S for which it is possible to set up an outside controller directly influencing the state of the nodes in S and strongly controlling the system.

Finding such a node set, called minimum-size input node set, for strong controllability is an NP-hard problem. To our best knowledge, there is currently no algorithm for systems with some particular structure. We partially solve this problem when the system is a tree.

In this paper, we solve the problem of determining a minimum-size input node set S for strong controllability whenever the system structure is a tree and either every node's state influences itself (self-damped systems) or every node's state is influenced by the state of some other nodes but not itself (undamped systems). To do so, we use the notion of zero forcing set, a concept borrowed from combinatorial matrix theory. Our method consists in finding a minimum zero forcing set in the simple tree associated with the system.

The outline of the paper is as follows: in Section II we provide some background used throughout the rest of the paper. The strong structural controllability of a system is presented in Section III. Section IV is devoted to the notion of zero forcing set and to their role in the study of the strong structural controllability of a system. Finally in Section V, we solve the problem of determining a minimum-size input node set for the strong controllability of a system with a self-damped/undamped tree structure. Section VI draws a short conclusion.

II. NOTATIONS AND PRELIMINARIES

In this section, we provide some background used in the rest of the paper.

An $n \times m$ zero-nonzero pattern (or *pattern* for short) is an $n \times m$ matrix whose each entry is either zero or a star.

A *realization* A of a pattern $\mathbf{A}$ is a real matrix whose any entry is nonzero if and only if the corresponding entry in $\mathbf{A}$ is a star. We write $A \in \mathbf{A}$.

Given an $n \times m$ pattern $\mathbf{A}$, we define the bipartite graph $B_{\mathbf{A}}$ as follows: the node sets of $B_{\mathbf{A}}$ are $V = \{1, ..., n\}$ and $V' = \{1', ..., m'\}$. Besides, $\{i, j'\}$ is an edge in $B_{\mathbf{A}}$ if and only if (i, j)-entry of $\mathbf{A}$ is a star.

A directed graph G on n nodes defines an $n \times n$ pattern $\mathbf{A}$ whose (i, j)-entry is a star if and only if there is a directed edge from node j to node i in G. The bipartite graph B_G associated with G is by definition the bipartite graph $B_{\mathbf{A}}$ associated with $\mathbf{A}$.

A t*-matching* in a bipartite graph is a set of t edges such that no two edges share a node. Given a matching, the nodes of the bipartite graph connected to an edge in the matching are called *matched* nodes, whereas the other nodes are called *unmatched* nodes. A t-matching is said to be *constraint* if there is no other t-matching with the same matched nodes.

By abuse of language, a constraint matching in the bipartite graph associated with a pattern $\mathbf{A}$ is referred as a constraint matching of $\mathbf{A}$.

Let T be a node subset of a directed graph G. A constraint T*-less* matching in the bipartite graph B_G is a constraint matching that does not contain edges of the form $\{i, i'\}$ with $i \in T$. In particular, if T contains all the nodes of G, then a constraint T-less matching is referred as a constraint *self-less* matching.

*This work was supported by the Belgian Programme of Interuniversity Attraction Poles and an Action de Recherche Concertée (ARC) of the Wallonia-Brussels Federation.

[1]ICTEAM Institute, Department of Applied Mathematics, Université catholique de Louvain, 1348 Louvain-La-Neuve, Belgium. maguy.trefois@uclouvain.be

[2]Center for Operations Research and Econometrics (CORE), Université catholique de Louvain, 1348 Louvain-La-Neuve, Belgium. jean-charles.delvenne@uclouvain.be

Given an $n \times m$ pattern $\mathbf{A}$ and a node set $S \subset \{1, ..., n\}$, $\mathbf{A}(S|.)$ is the pattern obtained from $\mathbf{A}$ by deleting the rows indexed by S.

Given a directed graph G, $G_\times$ denotes the graph obtained from G by putting a loop on each node of G. Similarly, the pattern obtained from a pattern $\mathbf{A}$ by putting stars along the diagonal of $\mathbf{A}$ is denoted $\mathbf{A}_\times$.

Given a graph G on n nodes and a node subset $S = \{i_1, ..., i_m\}$, the $n \times m$ pattern $B(S)$ is defined as

$$B(S) := [\mathbf{e}_{i_1}|\mathbf{e}_{i_2}|...|\mathbf{e}_{i_m}],$$

where any $\mathbf{e}_i$ is the $n \times 1$ pattern with a star in its i^{th} row and 0's otherwise.

III. Strong structural controllability

Consider a dynamical networked system represented by the directed graph G and consider an outside controller $u(t)$ directly influencing the state of the nodes in a node subset S. Such a set is called the input node set of the controlled system. The dynamics of the system is then described by the differential equation:

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

where A is a realization of the pattern $\mathbf{A}$ defined from G and B is a realization of the pattern $\mathbf{B}(S)$. This latter pattern has been defined so that only the nodes in S are controlled by the outside controller $u(t)$.

If the state matrix A is not completely known, Kalman's classical condition for controllability becomes inapplicable. Therefore, the structural controllability has been introduced. We distinguish two kinds of structural controllability: the weak one and the strong one. They provide respectively a lower and an upper bound on the minimum number of nodes that have to be controlled by the outside controller for classical control over the whole system.

Below, we present the strong structural controllability. For a description of the weak structural controllability and interesting references see [3].

Definition 3.1: The pair $(\mathbf{A}, \mathbf{B}(S))$ is strongly S-controllable if any realization (A, B) is controllable (in the sense that the controllability matrix is full rank).

The following test criterion for strong controllability was proved in [4]. Define V_s as the set of nodes with a loop in the original graph G. In terms of dynamics, this means that the state of each node in V_s influences itself.

Theorem 3.1: [4] Consider a system represented by a directed graph G on n nodes and a node set S with cardinality $m \leq n$. The pair $(\mathbf{A}, \mathbf{B}(S))$ is strongly S-controllable if and only if $\mathbf{A}(S|.)$ has a constraint $(n-m)$-matching and $\mathbf{A}_\times(S|.)$ has a constraint V_s-less $(n-m)$-matching.

Given a node set S, a $\mathcal{O}(n^2)$ algorithm was presented in [4] in order to test if the system is strongly S-controllable. If not, it provides an input node set $\tilde{S}$ containing S such that the system is strongly $\tilde{S}$-controllable.

Finding a minimum-size input node set for strong controllability is an NP-hard problem. However, this problem may be easy in the case of systems with some particular structure. The following theorem proved in [4] gives more insight about minimum-size input node sets for strong controllability of self-damped/undamped systems.

A *self-damped* graph is a graph with a loop on each node (every node's state influences itself), whereas an *undamped* graph is a loop-free graph (every node's state is influenced by the state of some other nodes but not itself).

Theorem 3.2: [4] Consider a system modeled by a directed graph G on n nodes which is self-damped or undamped. Consider a maximum constraint self-less $(n-m)$-matching in the bipartite graph (V, V', E) associated with $G_\times$, with unmatched nodes $S' \subset V'$ and $S \subset V$. Then S is a minimum-size input node set for strong controllability of the system.

This theorem provides thus a way to obtain a minimum-size input node set in the case of a self-damped/undamped system. However, computing a maximum constraint matching in a bipartite graph is known to be NP-hard [5].

In Section 5, we solve the problem of computing a minimum-size input node set when the system structure is a self-damped/undamped tree. To do so, we use the notion of zero forcing set. In the next section, we define this notion and we show how it is related to the strong controllability of the system.

IV. Zero forcing sets and strong controllability

This section is devoted to the notions of zero forcing number and zero forcing set and to the equivalence between the zero forcing sets in a directed graph G and the constraint matchings in the bipartite graph B_G associated with G. Thanks to this equivalence, we restate the results presented in the previous section about strong controllability in terms of zero forcing sets. These will be useful in the next section in order to determine a minimum-size input node set for the strong controllability of a system with a self-damped/undamped tree structure.

In order to present the zero forcing number/sets of a directed graph G, we need to define the following color change rule on the graph: suppose that any node of G is either black or white. If exactly one out-neighbor j of node i is white (possibly $j = i$), then change the color of j to black. Repeat this rule on each node of G until no more color change is possible.

When the color change rule is applied to a node i in order to change the color of node j, we say that i *forces* j and we write $i \rightarrow j$.

A *zero forcing set* of G is then a set of nodes in G such that if only these nodes are initially black, then after applying the color change rule repeatedly on G, the whole graph is black.

The *zero forcing number* of G, denoted $Z(G)$, is the minimum size of a zero forcing set in G.

A minimum zero forcing set is then a zero forcing set of size $Z(G)$ in G.

In order to state the theorem showing the equivalence between the zero forcing sets in G and the constraint matchings in the bipartite graph B_G, we need the definition of a chronological list of forces.

Given a zero forcing set of a directed graph G, we can list the forces in order in which they were performed in order to color the graph in black. Such a list is called a *chronological list of forces*.

Theorem 4.1: Let G be a directed graph and B_G the bipartite graph associated with G. Then, a node subset of G is a zero forcing set of G with a chronological list of forces $j_1 \rightarrow i_1, j_2 \rightarrow i_2, ..., j_t \rightarrow i_t$ if and only if $\mathcal{M} := \{\{i_1, j_1'\}, \{i_2, j_2'\}, ..., \{i_t, j_t'\}\}$ is a constraint matching in B_G.

Thanks to this equivalence, we can restate Theorems 3.1 and 3.2 as follows.

Theorem 4.2: Consider a system represented by a directed graph G on n nodes and a node set S with cardinality $m \leq n$. The pair $(\mathbf{A}, \mathbf{B}(S))$ is strongly S-controllable if and only if S is a zero forcing set of $G_\times$ for which there is a chronological list of forces that does not contain a force of the form $i \rightarrow i$ with $i \in V_s$ and a zero forcing set of G.

Theorem 4.3: Consider a system modeled by a directed graph G on n nodes which is self-damped or undamped. Consider a minimum zero forcing set S of $G_\times$ for which there is a chronological list of forces with no forces of the form $i \rightarrow i$. Then, S is a minimum-size input node set for the strong controllability of the system.

Thanks to these results, we completely solve the problem of determining a minimum-size input node set for strong controllability in the case of a system with a self-damped/undamped tree structure.

V. The case of the systems with a tree structure

In this section, we present an algorithm computing a minimum-size input node set for strong controllability of a system with a self-damped/undamped tree structure.

Systems with a tree structure appear in many different fields. For example, tree-structured organizations are designed in multiagent systems [7], and tree-structured stochastic processes are used in order to model complex data [8].

As previously, the term *directed graph* refers to a directed graph allowing loops. However, for the scope of our algorithm, we distinguish the directed graphs that allow loops from the simple directed graphs, that prohibit loops. In a simple directed graph G_s, the color change rule is slightly different: suppose that any node of G_s is either black or white. If node i is black and if node j is the only white out-neighbor of i, then change the color of j to black. Therefore, unlike in a directed graph allowing loops, in the case of a simple directed graph a node must be black to be able to force one of its out-neighbors.

Notice that a directed graph G with no loops can be considered either as a directed graph allowing loops or as a simple directed graph. According to the case, the color change rule on G is different.

Using this new color change rule, the zero forcing number and the zero forcing sets of a simple directed graph are defined as in the case of a directed graph allowing loops.

Consequently, notice the minimum zero forcing set S in Theorem 4.3 meets the definition of a minimum zero forcing set in the simple directed graph G_s associated with G, i.e. G_s is obtained from G by deleting its loops.

Corollary 5.1: Consider a system modeled by a self-damped/undamped directed graph G. Let S be a minimum zero forcing set in the simple graph G_s associated with G. Then, S is a minimum-size input node set for the strong controllability of the system.

The previous corollary claims that a minimum zero forcing set in the simple directed graph G_s is a minimum-size input node set for the strong controllability of G. However, computing a minimum zero forcing set in a simple directed graph is known to be NP-hard.

Below, we present a $\mathcal{O}(n^2)$ algorithm that computes a minimum zero forcing set in a simple tree on n nodes.

The term *tree* refers equivalently to an undirected tree or a symmetric directed tree, which is obtained from an undirected tree by replacing each edge $\{i, j\}$ by both directed edges $(i, j), (j, i)$. Trees allow loops.

The color change rules defined above are the same in the case of a (simple) directed graph as in an (simple) undirected graph: instead of considering the out-neighbors, the neighbors are taken into account.

With the algorithm presented below, the problem of computing a minimum-size input node set for strong controllability of a system with a self-damped/undamped tree structure is completely solved.

In order to present the algorithm, we need the following definition and proposition.

Consider a simple tree T and a node v of T. Then, $T - v$ denotes the forest obtained from T by removing node v as well as the edges connected to v. Denote $T_v^1, ..., T_v^k$ the connected components of $T - v$. If at least two of them are paths connected to v from one of their endpoints, then v is said to be an *appropriate* node.

Proposition 5.1: [6] Any tree on at least three nodes has an appropriate node.

Our algorithm is stated in Algorithm 1.

Since finding an appropriate node can be done in $\mathcal{O}(n)$ time from a depth-first search and since at each iteration at least three nodes are removed from G, our algorithm runs in $\mathcal{O}(n^2)$ time.

VI. Conclusion

Computing a minimum-size input node set for the strong controllability of a networked system is NP-hard. In this paper, we have solved this problem in the case of a self-damped or an undamped system with a tree structure. Our method is based on the notion of zero forcing set in a simple graph, a concept borrow from combinatorial matrix theory. For a tree on n nodes, our algorithm runs in $\mathcal{O}(n^2)$ time.

REFERENCES

[1] C.-T. Lin, Structural controllability, IEEE Trans. on Automatic Control, vol. AC-19, no. 3, pp. 201-208, Aug. 1974.

[2] H. Mayeda, T. Yamada, Strong structural controllability, SIAM Journal on Control and Optimization , vol. 17, no. 1, pp. 123-138, 1979.

[3] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabasi, Controllability of complex networks, Nature, vol. 473, no. 7346, pp. 167-173, May 2011.

[4] A. Chapman, M. Mesbahi, Strong structural controllability of networked dynamics, 6141-6146, in Proc. American Control Conference (2013).

[5] M.C. Golumbic, T. Hirst, M. Lewenstein, Uniquely Restricted Matchings, Algorithmica, 31(2001), no. 2, pp. 139-154.

[6] I.-J. Kim, Algorithmic proof of Maxmult(T) = $p(T)$, Commun. Korean Math. Soc. 27(2012), No.4, pp. 665-668.

[7] Y.-P. So, E.H. Durfee, Designing Tree-Structured Organizations for Computational Agents, Computational and Mathematical Organization Theory 2:3 (1996): 219-246.

[8] J. Saunderson, V. Chandrasekaran, P.A. Parrilo, A.S. Willsky, Tree-Structured Statistical Modeling via Convex Optimization, Conference on Decision and Control and European Control Conference, pp. 2883 - 2888, 2011.

Algorithm 1:

Input: a simple tree T;

Output: a minimum zero forcing set S of T;

Set $S = \emptyset$;

$G = T$;

While G is non empty

- if G is a forest whose each connected component has less than 3 nodes, then put a node of each component in S and set $G = \emptyset$;
- else consider an appropriate node v of G in a connected component T_i of G;
  - denote $k(\geq 2)$ the number of connected components of $T_i - v$ which are paths connected to v from an endpoint;
  - for $k-1$ among them, put in S the endpoint which is a leaf in T_i;
  - remove v from G as well as the k paths connected to v from an endpoint;

end