On State-Dependent Dynamic Graphs and Their Controllability Properties

Mehran Mesbahi

***Abstract*— We consider distributed dynamic systems operating over a graph or a network. The geometry of the network is assumed to be a function of the underling system's states- giving it a unique dynamic character. Certain aspects of the resulting abstract structure, having a mixture of combinatorial and system theoretic features, are then studied. In this venue, we will explore an interplay between notions from extremal graph theory and system theory by considering a controllability framework for such state-dependent dynamic graphs.**

***Index Terms*— State-dependent graphs, dynamic graphs, graph controllability, Szemerédi's regularity.**

I. Introduction

This paper is motivated by system theoretic problems in the area of control and coordination of distributed dynamic systems operating over an information network. Many examples of such systems can be found in the literature, e.g., [2], [8], [11], [13], [15], [19]. Our own point of view in studying these problems has been influenced by the field of distributed space systems such as the Terrestrial Planet Imager (TPI) mission [19]. For a class of such systems (e.g., TPI) it is often the case that in order to control *over* the information network, it becomes necessary to have a priori control of the network itself. This is the result of an intricate interplay between the distributed system state on one hand and the properties of the underlying network on the other. The aspect of network control problem that we henceforth study is obtained by tying in the network attributes to the dynamic properties of the elements residing at each node. Specifically, we consider a scenario where the *existence* of an information channel between a pair of dynamic elements is determined, either fully or partially, by their respective states. We call the resulting structure- having a blend of dynamic and combinatorial features- a state-dependent dynamic graph. As we will see in this paper, state-dependent dynamic graphs not only give the network control problem a rather system theoretic flavor but also invite us to consider a host of new problems in control theory that are distinctively combinatorial [10]. One such problem introduced and studied in the present paper pertains to the controllability of such state-dependent dynamic graphs.

A. Two illustrative examples

Consider a set of cubical dynamic elements with color coded sides. Let us assume that each color represents one type of sensing or communication device that the element employs to exchange information with the other elements in the system. Moreover, we assume that each pair can exchange information if the correct colored sides are facing each other. Evidently, as the orientations of these cubical elements evolve in time, we obtain a sequence of information graphs; in particular, we realize that the information graph is state-dependent and in general, dynamic. Another such example, of particular relevance in the control of multiple unmanned aerial vehicles (UAVs), is the nearest neighbor information exchange paradigm [11]. In this framework, there is an information channel (e.g., relative sensing) between a pair of UAVs if they are within a given distance of each other. As the positions of the UAVs evolve in time, say during the course of a reconfiguration, the underlying information exchange infrastructure naturally evolves in time as well. Again, in the nearest neighbor information exchange scenario, one has to consider information graphs that are inherently state-dependent, and thereby, dynamic.

B. Notation

Our graph theoretic terminology is standard [4], [6]. We take for granted the following notions: a (simple, finite) graph $G = (V, E)$, subgraphs ($\widehat{G} \subseteq G$ if its vertex and edge sets are subsets of those of G), induced subgraphs ($\widehat{G} \subseteq_I G$ if $\widehat{G}$ is a subgraph of G and contains all of G's potential edges), the order of graph G, $|G|$ (cardinality of its vertex set), and its size $\|G\| = |E|$ (cardinality of its edge set), incidence relation (edge $e \in E$ is incident to $u, v \in V$ if $e = \{u, v\}$; u and v are endpoints of the edge $\{u, v\}$; edge e is incident on vertex v if v is one of e's endpoints), degree of a vertex (the number of edges incident on it), labeling of graphs, isomorphic graphs, connected, complete, and empty graphs. K_n will denote the complete graph of order n; its complement, $\overline{K}_n$ will be the empty graph (of order n and size zero). If X_i and X_j are two disjoint sets of vertices of G, we denote by $e(X_i, X_j)$ the set of edges of G with one endpoint in X_i and the other endpoint in X_j; the cardinality of this set will be denoted by $\|X_i, X_j\|$. The set of all labeled graphs of order N will be designated by $\mathcal{G}_N$; finally $\{1, \ldots, N\}$ will be denoted by $[N]$.

C. Related Works

State-dependent graphs have not apparently been studied in the framework proposed in this paper; there are however a few closely related research works. First, we mention the work of Aizerman *et al.* [1],[1] where a process of the form $G(t+1) = F(G(t))$ is considered- with $G(t)$ denoting the graph structure at time t, and F a transformation that maps this graph to $G(t+1)$. As the authors of [1] point out, the motivation for their work had come from mathematical studies on administrative structures, organization of communication and service systems, arrangement for the associative memory of a computer, etc. In [1] , after introducing the notion of "subordinate" functions that operate over trees, the authors consider fixed point and convergence properties of the resulting graph transformation. Related to this work, and in particular to the preceding paper of Petrov [16], the notions of web and graph grammars are also aligned with our state-dependent graphs. Graph grammars are composed of a finite

[1] Of "Aizerman's conjecture" fame.

July 2004; this material is based upon work supported by National Science Foundation under Grant No. 0301753.

The author is with the Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195-2400 USA (email: mesbahi@aa.washington.edu)

set of initial graphs, equipped with and closed with respect to a finite set of rules, for their local graph transformation. The general area of graph grammars has flourished as an active area of research in computer science, with many applications in software specification and development, computer graphics, database design, and concurrent systems [18].

Another area of research related to the present work is graph dynamics as described in [17] and references therein. The motivation for this line of work comes from an attempt to generalize a wide array of results in graph theory pertaining to the line graph operator. The line graph of a graph highlights the incidence relation among the *edges* of the original graph (two edges are incident if they have a common vertex). The line graph operator- its inverse and fixed points- have historically been of great interest in the combinatorics community. An early result in the field goes back to early nineteen thirties, where Whitney showed that every finite connected graph, except the triangle, has at most one connected line graph inverse [20]. Over the ensuing years, the research on graph dynamics has evolved along similar dynamical system-type questions, pertaining to characterization of convergence properties, fixed points, and inverses, of various classes of graph transformations.

II. Basic Setup

Our distributed system consists of N indexed dynamic elements $1, 2, \ldots, N$. We view the system as a labeled graph on N vertices with each vertex being a (controlled) dynamical system. The state of vertex i will be denoted by $x_i \in X_i$. Although the specific nature of these dynamic elements is not of central importance in this paper, we have chosen to work with discrete-event finite state space systems that are endowed with a global clock. In particular, we consider systems with a finite state space $X_i = \{1, \ldots, n_i\} = [n_i]$ and finite control space $U_i = \{1, \ldots, m_i\} = [m_i]$, with state evolution described by

$$x_i(k+1) = x_i(k) \oplus u_i, \tag{2.1}$$

expressing the value assumed by the state at $k+1$ when the system is initialized at $x_i(k)$ and the event u_i occurs; if no event occurs, $x_i(k+1) = x_i(k)$. The state space of such a system assumes the form

$$X := X_1 \times X_2 \times \ldots \times X_N; \tag{2.2}$$

$x \in X$, the juxtaposed vector of x_i's, is referred to as the state of the distributed system.

A. State-dependent graphs

A state-dependent graph is a mapping, $g_{\mathcal{S}}$, from the distributed system state space X (2.2), to the set of all labeled graphs of order N, $\mathcal{G}_N$, i.e.,

$$g_{\mathcal{S}} : X \to \mathcal{G}_N \quad \text{and} \quad g_{\mathcal{S}}(x) = G; \tag{2.3}$$

we will occasionally write $G_x := g_{\mathcal{S}}(x)$ to highlight the dependency of the resulting graph on the state. It is assumed that the order of these graphs, N, is fixed; their edge set, $E(g_{\mathcal{S}}(x))$, however is a function of state x. We need to further specify *how* the state of the system dictates the existence of an edge between a pair of vertices in the state-dependent graph. This is achieved by considering the subset $S_{ij} \subseteq X_i \times X_j$ and requiring

$$ij \in E(g_{\mathcal{S}}(x)) \quad \text{if and only if} \quad (x_i, x_j) \in S_{ij};$$

we call S_{ij} the edge-states of vertices i and j.[2] As an example, for the nearest neighboring scenario of §I.A, the edge-states are of the form

$$S_{ij} = \{(\begin{bmatrix} r_i \\ v_i \end{bmatrix}, \begin{bmatrix} r_j \\ v_j \end{bmatrix}) \,|\, \|r_i - r_j\| \leq \rho\},\ i, j \in [N],\ i \neq j,$$

where r_i and v_i represent, respectively, the position and the velocity of UAV i, and ρ is a given positive number. In general, we will denote the collection of the edge-states S_{ij} by

$$\mathcal{S} := \{S_{ij}\}_{i,j \in [N], i \neq j} \quad \text{with} \quad S_{ij} \subseteq X_i \times X_j. \tag{2.4}$$

Example 2.1: Consider two square elements i and j, the four sides of which have alternatively been labeled by 0 and 1. The state of each square, x, is thus represented by one binary state, interpreted as the label that is facing "up." Consider the scenario where there is an edge between the vertices i and j if $x_i + x_j = 0 \pmod 2$. The state space partitions are therefore $x_i(1) = 0$, $x_i(2) = 1$, $x_j(1) = 0$, $x_j(2) = 1$, and the set of edge-states is identified as $S_{ij} = \{(0, 0), (1, 1)\}$.

Definition 2.2: Given the set system $\mathcal{S}$ (2.4), we call the map $g_{\mathcal{S}} : X \to \mathcal{G}_N$ with an image consisting of graphs of order N, having an edge between vertex i and j if and only if $(x_i, x_j) \in S_{ij}$, a state-dependent graph with respect to $\mathcal{S}$.[3]

The image of the state-dependent graph $g_{\mathcal{S}}$,

$$\{G \,|\, g_{\mathcal{S}}(x) = G,\ \text{for some}\ x \in X\} = \{G_x \,|\, x \in X\},$$

will be denoted by $g_{\mathcal{S}}(X)$.

III. Graphical Equations

It seems natural that prior to studying dynamic state-dependent graphs, resembling systems that are described by differential equations, static state-dependent graphs are examined first. In this section we hence consider problems related to checking for the existence of, and possibly solving for, the states that have a particular graph realization.

A. System of inequalities and equation solving for specific graphs

Given the set $\mathcal{S}$ (2.4) and a labeled graph G of order N, we consider finding solutions to the equation

$$g_{\mathcal{S}}(x) = G. \tag{3.5}$$

Note that depending on specific applications, equality between a pair of graphs in (3.5) can be considered as a strict equality or up to an isomorphism. In fact in § IV, we will consider

[2] Although the existence of an edge between two elements can potentially depend on the states of other elements, we will not consider this case in the paper. Such more general state-dependency schemes would lead us to state-dependent hypergraphs and set systems [3].

[3] We could alternatively call $\mathcal{S}$ (2.4) itself the state-dependent graph.

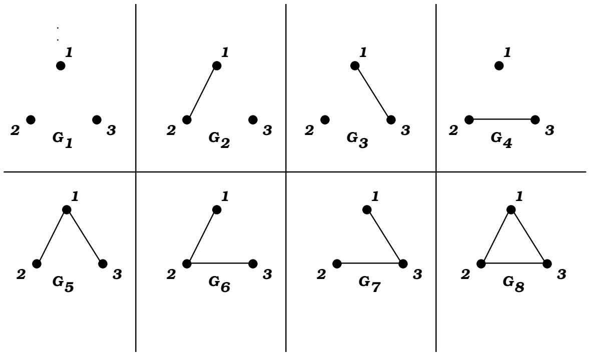

Fig. 1. Labeled graphs on three vertices

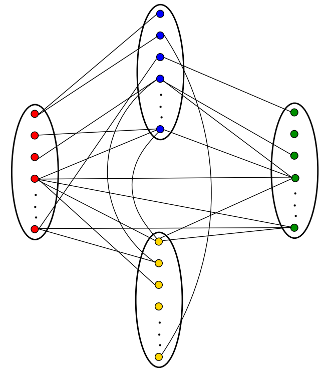

Fig. 2. The supergraph $\mathcal{G}(N, \mathcal{S})$

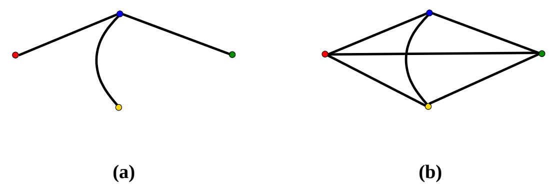

Fig. 3. (a) A subgraph of the supergraph $\mathcal{G}(N, S)$ in Figure 2, (b) an induced subgraph of $\mathcal{G}(N, S)$

a scenario where it is more natural to consider a subgraph inclusion relation of the form $g_{\mathcal{S}}(x) \subseteq G$ rather than (3.5).

Graphical equation solving can become equivalent to solving systems of equations and inequalities, depending on the characterization of the edge-states S_{ij}'s in $\mathcal{S}$ (2.4). Let us elaborate on this observation via two examples.

Example 3.1: Let $G = (V, E)$, $V = \{1, 2, 3\}$. The eight possible labeled graphs on three vertices is shown in Figure 1, with G_8 denoting K_3. Thus the equation $g_{\mathcal{S}}(x) = G_8$ has a solution if and only if the set $g^{-1}(G_8) = \{x \,|\, (x_1, x_2) \in S_{12} \;\&\; (x_1, x_3) \in S_{13} \;\&\; (x_2, x_3) \in S_{23}\}$ is nonempty. Similarly, the equation $g_{\mathcal{S}}(x) = G_6$, is solvable if and only if the set $g^{-1}(G_6) = \{x \,|\, (x_1, x_2) \in S_{12} \;\&\; (x_1, x_3) \in \overline{S}_{13} \;\&\; (x_2, x_3) \in S_{23}\}$ is nonempty.

Example 3.2: For all $i, j \in [3]$, $i \neq j$, let us assume that

$$S_{ij} = \{(x_i, x_j) \,|\, q_{ij}(x_i, x_j) \leq 0\} \subseteq \mathbb{R}^n \times \mathbb{R}^n,$$

where $q_{ij}(x_i, x_j) = x_i^T Q_{ij} x_j$ and $Q_{ij} = Q_{ij}^T \in \mathbb{R}^{n \times n}$. Then the equation $g_{\mathcal{S}}(x) = G_5$ (Figure 1) has a solution if and only if the set $\{x \,|\, q_{12}(x_1, x_2) \leq 0, q_{13}(x_1, x_2) \leq 0, q_{23}(x_2, x_3) > 0\}$ is nonempty. On the other hand, this set is empty if and only if $q_{12}(x_1, x_2) \leq 0$ and $q_{13}(x_1, x_3) \leq 0$ imply that $q_{23}(x_2, x_3) \leq 0$. Now extend each q_{ij}, making them functions of $x = [x_1, x_2, x_3]^T$, e.g., $\widetilde{q}_{ij}(x) := q_{ij}(x_i, x_j) + 0 \times x_k$ where $k \neq i$ and $k \neq j$, and let $\widetilde{q}_{ij}(x) = x^T \widetilde{Q}_{ij} x$. Using the S-procedure [5], we conclude that $g_{\mathcal{S}}(x) = G_5$ has no solution if there exist $\tau_1, \tau_2 \geq 0$ such that $\widetilde{Q}_{23} \leq \tau_1 \widetilde{Q}_{12} + \tau_2 \widetilde{Q}_{13}$, where an inequality between two symmetric matrices is interpreted in terms of the ordering induced by positive semi-definite matrices.

B. Supergraphs and equation solving for a class of graphs

For each pair of dynamic elements- in our case, finite automata with a finite state-space- the edge-states can conveniently be represented by a bipartite graph. In this section we point out a useful characterization of state-dependent graphs in view of this equivalence.

Define the supergraph $\mathcal{G}(N, \mathcal{S})$ as the union of bipartite graphs, each representing the edge-states between a pair of dynamic elements (Figure 2). The supergraph thus consists of N "supervertices" representing the states of the elements in the distributed system. A vertex that represents the particular state of a specific element will be referred to as a "subvertex." Note that subgraphs of the supergraph $\mathcal{G}(N, \mathcal{S})$, employing one subvertex from each supervertex, are exactly those graphs that can be assumed by the distributed system. We call the graph G a transversal (subgraph) of the supergraph $\mathcal{G}(N, \mathcal{S})$ if (1) it is an order N subgraph of $\mathcal{G}(N, \mathcal{S})$ and (2) each of its N vertices belong to distinct supervertices of $\mathcal{G}(N, \mathcal{S})$. If G is a transversal subgraph of $\mathcal{G}(N, \mathcal{S})$ and contains all of its potential edges, it is called an induced transversal. We will denote the subgraph and induced subgraph transversals as in $G \subseteq_T \mathcal{G}(N, \mathcal{S})$ and $G \subseteq_I \mathcal{G}(N, \mathcal{S})$, respectively. The supergraph construction has the following immediate ramification.

Proposition 3.3: Given the collection of edge-states $\mathcal{S}$ (2.4), the equation $g_{\mathcal{S}}(x) = G$ has a solution if and only if $G \subseteq_I \mathcal{G}(N, \mathcal{S})$. Furthermore, the inclusion $G \subseteq g_{\mathcal{S}}(x)$ has a solution if and only if $G \subseteq_T \mathcal{G}(N, \mathcal{S})$.

IV. Dynamic Graph Controllability

In this section, we consider a controllability concept for state-dependent dynamic graphs. Controllable (state-dependent) dynamic graphs are processes where every graph (either labeled or up to an isomorphism, depending on the context) is reachable by a judicious choice of the control sequence. In this venue, it is natural to explore connections between the controllability of the distributed system state, referred henceforth as the x-process, and the corresponding graph process, which we will refer to as the G-process.[4] It is particularly of interest to identify a scenario where the controllability of the x and the G-processes are in a sense equivalent. Let us denote by $\mathcal{G}_N^{\Delta}$ the set of graphs of order

[4]We will not delve into the various notions of controllability: the x-process is controllable if for two arbitrary states, there exists a control sequence that steers one to the other.

N with maximum vertex degree Δ; thus $\mathcal{G}_N^0 = \{\overline{K}_N\}$ and $\mathcal{G}_N^{N-1} = \mathcal{G}_N$. Our first notion for graph controllability relies on graph reachability- as a subgraph- from an arbitrary initial state.

Definition 4.1: The G-process is strictly Δ-controllable if for any $G_f \in \mathcal{G}_N^\Delta$ and any initial state $x_0 \in X$, there exist a finite k and an x-process for which $G_f \subseteq g_\mathcal{S}(x(k))$. When $\Delta = N-1$, we refer to strictly Δ-controllable G-processes as strictly controllable.

Including the maximum degree Δ qualification in the definition of graph controllability is not purely accidental, although other graph parameters could be employed in this definition as well. The maximum vertex degree however does have implications on the relative sensing overhead of a distributed system and the overall graph "complexity," and as such, is a prudent graph parameter in Definition 4.1.

Our second definition of graph controllability relies more on the properties of the G-process itself rather than the x-process.

Definition 4.2: The G-process is Δ-controllable if for two graphs $G_0, G_f \in \mathcal{G}_N^\Delta$, there exist a finite k and an x-process, for which $G_0 \subseteq g_\mathcal{S}(x(0))$ and $G_f \subseteq g_\mathcal{S}(x(k))$. When $\Delta = N-1$, we refer to a Δ-controllable G-processes simply as controllable.

The distinction between Definitions 4.1 and 4.2 is the qualification on graph reachability from an arbitrary state versus an arbitrary graph- the latter being less stringent. Observe that Δ-controllability has a cascading property, as Δ_0-controllability for some $\Delta_0 > 0$, implies Δ-controllability for all $\Delta \geq \Delta_0$. In order to highlight the connection between strict controllability and controllability of the G-process, we introduce the notion of calmness.

Definition 4.3: A graph $G \in \mathcal{G}_N^\Delta$ is strictly calm with respect to the controlled x-process, if (1) for any x_0, x_f for which $G \subseteq g_\mathcal{S}(x_0), g_\mathcal{S}(x_f)$, there exists a control sequence that steers x_0 to x_f, and (2) for all intermediate states $G \subseteq g_\mathcal{S}(x)$.

Hence the empty graph $\overline{K}_N$ is always strictly calm. When the second qualification in the above definition fails to hold, the graph $G \in \mathcal{G}_N^\Delta$ is simply called calm. Moreover, when every element of a subset of $\mathcal{G}_N^\Delta$ is (strictly) calm with respect to the x-process, the subset itself is referred to as (strictly) calm. Let us denote by

$$\begin{aligned} X_\Delta &:= \{x \in X \mid \max\deg g_\mathcal{S}(x) \leq \Delta\} \\ &= \{x \in X \mid G_x \in g_\mathcal{S}(X) \cap \mathcal{G}_N^\Delta\}. \end{aligned}$$

Proposition 4.4: If the G-process is Δ-controllable and $g_\mathcal{S}(X)$ is calm with respect to the x-process, then the G-process is strictly Δ-controllable in X_Δ.

Proof: Let $z \in X_\Delta$ and $G_f \in \mathcal{G}_N^\Delta$ be given. It suffices to show that there is a control sequence steering z to x_f where $G_f \subseteq g_\mathcal{S}(x_f)$. The Δ-controllability of the G-process does imply that there is an x-process ($x_0 \rightsquigarrow x_f$) for which $g_\mathcal{S}(z) \subseteq g_\mathcal{S}(x_0)$ and $G_f \subseteq g_\mathcal{S}(x_f)$. As $g_\mathcal{S}(z), g_\mathcal{S}(x_0) \subseteq g_\mathcal{S}(x_0)$ and $g_\mathcal{S}(x_0)$ is Δ-calm, there is a control sequence that steers z to x_0. Joining the two control sequences together now completes the proof. ∎

We now proceed to explore the controllability correspondence between the x and the G-processes. First note that if there is a natural bijection between the x and the G processes, then their system theoretic properties will have a more direct interrelationship with each other. Let us digress a bit to introduce a notion that essentially allows us to "invert" the edge-state dependency map $g_\mathcal{S}$ (2.3).

A. *Szemerédi's Regularity*[5]

Given the collection of edge-states $\mathcal{S}$ (2.4), the edge-state density between elements i and j is defined as

$$d_\mathcal{S}(X_i, X_j) = \frac{\|X_i, X_j\|}{|X_i|\,|X_j|}. \tag{4.6}$$

In this section we will assume, without loss of generality, that $\|X_i\| = n$ for all i, and thus $d_\mathcal{S}(X_i, X_j) = |S_{ij}|/n^2$. The $n(n-1)/2$ numbers $d_\mathcal{S}(X_i, X_j)$ (4.6), each ranging from zero and one, reflect the ratios of the states that are designated as the edge-states.

Definition 4.5 ([12]): For $\epsilon > 0$, the pair (X_i, X_j) is called ϵ-regular at level ρ if (1) $d_\mathcal{S}(X_i, X_j) \geq \rho$ and (2) for every $Y_i \subseteq X_i$ and $Y_j \subseteq X_j$ satisfying $|Y_i| > \epsilon\,|X_i|$ and $|Y_j| > \epsilon\,|X_j|$ one has

$$|d_\mathcal{S}(X_i, X_j) - d_\mathcal{S}(Y_i, Y_j)| < \epsilon. \tag{4.7}$$

The supergraph $\mathcal{G}(N, \mathcal{S})$ (see §III.B) is ϵ-regular at level ρ if each pair of its supervertices is ϵ-regular at a level at least ρ. We will denote an ϵ-regular supergraph at level ρ by $\mathcal{G}(N, \mathcal{S}_{\epsilon,\rho})$ (recall that each supervertex has n subvertices). One of the most important consequences of regularity is the following observation [12].

Proposition 4.6: Consider two ϵ-regular supervertices X_i, X_j with $d(X_i, X_j) = \rho$. Let $\Psi \subseteq X_i$ be the set of all subvertices with at least $(\rho - \epsilon)\,|X_j|$ neighbors in X_j. Then $|\Psi| \geq (1-\epsilon)\,|X_i|$.

Proof: Suppose that the number of elements $x_i \in X_i$, having strictly less neighbors in X_j than $(\rho-\epsilon)\,|X_j|$, is strictly more than $\epsilon\,|X_i|$. Denote this set by $\overline{\Psi}$. Then $\|\overline{\Psi}, X_j\| < (\rho - \epsilon)\,|\overline{\Psi}|\,|X_j|$, i.e., $d(\overline{\Psi}, X_j) < (\rho-\epsilon)$ and the pair $(\overline{\Psi}, X_j)$ violates the regularity assumption on the pair (X_i, X_j) (4.7). ∎

B. *Key Lemma*

The regularity of the supergraph $\mathcal{G}(N, \mathcal{S})$ provides a level of transparency between the x-process and the G-process, allowing us to make a correspondence between their controllability properties. Such an "invertibility" condition for the map $g_\mathcal{S}$ (2.3) will now be made more explicit in terms what is commonly known as the Key Lemma [12].

Lemma 4.7: Let $\rho > \epsilon > 0$ be given. Consider an ϵ-regular supergraph $\mathcal{G}(N, \mathcal{S}_{\epsilon,\rho})$ and let $\delta := \rho - \epsilon$. Let H be a graph of order N with maximum vertex degree $\Delta(H) > 0$. If $\delta^{\Delta(H)}/(1+\Delta(H)) \geq \epsilon$, then $H \subseteq_T \mathcal{G}(N, \mathcal{S}_{\epsilon,\rho})$; moreover, the number of such H-subgraph transversals is at least

$$(\delta^{\Delta(H)} - \Delta(H)\,\epsilon)^N\, n^N. \tag{4.8}$$

[5] Regularity Lemma was originally employed in combinatorial number theory to resolve a famous conjecture of Erdős and Turán [7]. Its application to some open problems in extremal graph theory is more recent; the reader is referred to the excellent survey of Komlós and Simonovits [12].

Let us denote by $\Gamma(x)$ the set of neighboring subvertices of x in the supergraph $\mathcal{G}(N,\mathcal{S})$. The Key Lemma 4.7 has the following constructive proof.

Embedding Algorithm:

Initialize the sets $C_{0,j} = X_j$ for all $j = 1,\ldots,N$ and set $i = 1$.

1) Pick $x_i \in C_{i-1,i}$, such that for all $j > i$ for which $(i,j) \in E(H)$, one has
$$|\Gamma(x_i) \cap C_{i-1,j}| > \delta\,|C_{i-1,j}|; \qquad (4.9)$$
Proposition 4.6 guarantees that the set of such states is nonempty; in fact, the cardinality of the set of states that *violate* (4.9) is at most $\Delta(H)\,\epsilon\,n$.
2) For each $j > i$, let
$$C_{i,j} = \begin{cases} \Gamma(x_i) \cap C_{i-1,j} & \text{if } (i,j) \in E(H) \\ C_{i-1,j} & \text{otherwise.} \end{cases}$$
3) If $i = N$, terminate the algorithm; otherwise, let $i = i+1$ and go to Step 1.

In Step 2 of the algorithm, denote the cardinality of the set $\{k \in [i]\,|\,(k,j) \in E(H)\}$ by d_{ij}; then one has $|C_{i,j}| > \delta^{d_{ij}}\,n \geq \delta^{\Delta(H)}\,n$, when $d_{ij} > 0$, and $|C_{ij}| = n$ when $d_{ij} = 0$. In both cases, $|C_{i,j}| > \delta^{\Delta(H)}\,n$ when $j > i$. Moreover, when choosing the exact location of x_i all but at most $\Delta(H)\epsilon n$ vertices of $C_{i-1,i}$ satisfy (4.9) as needed in Step 1 of the algorithm. Thus when finding the transversal H in $\mathcal{G}(N,\mathcal{S}_{\rho,\epsilon})$, at least
$$|C_{i-1,i}| - \Delta(H)\,\epsilon\,n > \delta^{\Delta(H)}\,n - \Delta(H)\,\epsilon\,n \qquad (4.10)$$
free choices exist for each x_i. The estimate (4.8) for the number of embeddings of H in $\mathcal{G}(N,\mathcal{S}_{\rho,\epsilon})$ now follows from (4.10).

Example 4.8: In the network of N elements, each with n states, $x_i \in X_i$ is called a blind state of i with respect to j if $(x_i,x_j) \notin S_{ij}$ for all $x_j \in X_j$; denote by b_{ij} the number of such states. For this example, let $b_{ij} = b_{ji}$ and furthermore, assume that all other states are edge states, i.e.,
$$S_{ij} = (X_i \times X_j) \setminus (\{i\text{-th element's blind states w.r.t. } j\} \times \{j\text{-th element's blind states w.r.t. } i\}).$$
Thus $\rho := d_{\mathcal{S}}(X_i,X_j) = 1 - (b_{ij}^2/n^2)$. We now proceed to check for the existence of subgraphs with vertex degree of at most 2 for the corresponding state-dependent graph on N elements. Let $\epsilon = m/n$. Lemma 4.7 suggests that we need to ensure the inequality
$$\rho \geq \epsilon + \sqrt{3}\,\epsilon, \qquad (4.11)$$
and that for all $Y_i \subseteq X_i$ and $Y_j \subseteq X_j$ of size strictly greater than m,
$$|\,\rho - d_{\mathcal{S}}(Y_i,Y_j)\,| < \frac{m}{n}. \qquad (4.12)$$
The maximum deviation of the quantity $d_{\mathcal{S}}(Y_i,Y_j)$ from the edge state density ρ occurs when
$$d_{\mathcal{S}}(Y_i,Y_j) = 1 - \frac{b_{ij}^2}{(m+1)^2}.$$

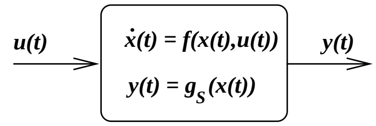

Fig. 4. Dynamics of state-dependent graphs

Thus for ϵ-regularity it suffices to have
$$\frac{n}{m(m+1)^2} - \frac{1}{mn} < \frac{1}{b_{ij}^2}. \qquad (4.13)$$
We now note that for particular values of n and m in (4.11) and (4.13), one can obtain an allowable number of blind states between each pair of elements so that state-dependent subgraphs with vertex degree of at most 2 are guaranteed to exist for the N-element network. For example, when $n = 100$ and $m = 20$, having $b = 9$ satisfies both inequalities (4.11) and (4.13). Thereby, almost ten percent of each element's states can be blind states with respect to another element, while still guaranteeing the existence of any state-dependent subgraph with a maximum vertex degree 2. In fact, the bound (4.8) indicates that there are plenty of such subgraph transversals in the corresponding supergraph- in this example, at least 20^N of them!

Few remarks are in order at this point. Note that guaranteeing the existence of a transversal embedding H in the supergraph $\mathcal{G}(N,\mathcal{S}_{\epsilon,\rho})$ does not depend on the parameters ρ or ϵ individually. In fact, it is their difference $\rho - \epsilon$ that dictates the number of such embeddings in $\mathcal{G}(N,\mathcal{S}_{\epsilon,\rho})$, i.e., it is the *relative* order of density with respect to the "fineness" of regularity that prescribes the number of embeddings. In the meantime, the maximum vertex degree of the desired embedding accounts for the ease by which it can be embedded in the supergraph $\mathcal{G}(N,\mathcal{S}_{\epsilon,\rho})$, that is, to be realized by a judicious choice of the underlying dynamic states. Furthermore, the embedding algorithm suggests a constructive approach through which the desired subgraph can be synthesized.

C. Graph Controllability

We now reach the main result of this section- stated for the finite state automata case, but generalizable to other classes of dynamic systems (see Figure 4).[6]

Theorem 4.9: The G-process is Δ-controllable if the x-process is controllable and the supergraph $\mathcal{G}(N,\mathcal{S}_{\epsilon,\rho})$ satisfies $(\rho-\epsilon)^{\Delta}/(1+\Delta) \geq \epsilon$. On the other hand, the x-process is controllable if the G-process is controllable and $g_{\mathcal{S}}(X)$ is calm with respect to the x-process.

Proof: Assuming that the x-process is controllable, consider graphs $G_0, G_f \in \mathcal{G}_N^{\Delta}$. By regularity of the supergraph $\mathcal{G}(N,\mathcal{S}_{\epsilon,\rho})$, there exist $x_0, x_f \in X$ such that $G_0 \subseteq g_{\mathcal{S}}(x_0)$ and $G_f \subseteq g_{\mathcal{S}}(x_f)$. By controllability of the x-process however, there is a control sequence that steers x_0 to x_f; thus the G-process is Δ-controllable. Now assume the controllability of

[6]The generalization involves partitioning the state-space to finitely many regions and employing "measure" in place of "size" to obtain the required extension for the notion of regularity.

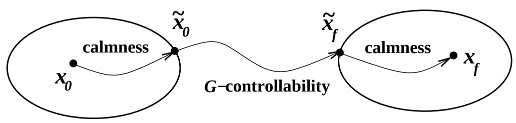

Fig. 5. Controllability of the x-process in Theorem 4.9

the G-process and consider an arbitrary pair $x_0, x_f \in X$, with the corresponding $g_{\mathcal{S}}(x_0), g_{\mathcal{S}}(x_f) \in \mathcal{G}_N$. Thereby, there are $\widetilde{x}_0$, $\widetilde{x}_f$, and a control sequence such that $g_{\mathcal{S}}(x_0) \subseteq g_{\mathcal{S}}(\widetilde{x}_0)$ and $g_{\mathcal{S}}(x_f) \subseteq g_{\mathcal{S}}(\widetilde{x}_f)$. As $g_{\mathcal{S}}(\widetilde{x}_0) \subseteq g_{\mathcal{S}}(\widetilde{x}_0)$ and $g_{\mathcal{S}}(\widetilde{x}_f) \subseteq g_{\mathcal{S}}(\widetilde{x}_f)$, by the calmness assumption (Definition 4.3), there is a control sequence from x_0 to $\widetilde{x}_f$, and analogously from $\widetilde{x}_f$ to x_f (see Figure 5). By joining these three control sequences together, we obtain a control sequence that steers x_0 to x_f, and hence the controllability of the x-process. ■

Hence, when the underlying x-process in Example 4.8 is controllable, the associated G-process is ensured to be 2-controllable.

Concluding Remarks

Motivated by a class of problems associated with control of distributed dynamic systems, in this paper we considered graphs with incidence relations that are dictated by the underlying dynamic states. We subsequently considered solving graphical equations over such state-dependent graphs, followed by introducing a controllability concept for the corresponding dynamic graphs. This work points to a new research direction in system and control theory that has a distinctive combinatorial character- potentially having a range of elegant connections with extremal combinatorics and graph theory.

Acknowledgments

The author gratefully acknowledges comments and suggestions by the anonymous referees.

References

[1] M. A. Aizerman, L. A. Gusev, S. V. Petrov, I. M. Smirnova, and L. A. Tenenbaum. Dynamic approach to analysis of structures described by graphs (foundations of graph-dynamics). In *Topics in the General Theory of Structures*, Edited by E. R. Caianiello and M. A. Aizerman. D. Reidel Publishing Company, Dordrecht, Holland, 1987.

[2] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control, *IEEE Transactions on Control Systems Technology*, (9) 6: 777–790.

[3] B. Bollobás. *Combinatorics: Set Systems, Hypergraphs, Families of Vectors, and Combinatorial Probability*, Cambridge, 1986.

[4] B. Bollobás. *Modern Graph Theory*. Springer-Verlag, 1998.

[5] S. P. Boyd, L. EL Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, 1994.

[6] R. Diestel. *Graph Theory*. Springer-Verlag, 2000.

[7] P. Erdős and P. Turán. On some sequences of integers, *Journal of London Mathematical Society*, 11: 261–264, 1936.

[8] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations, *IEEE Transactions on Automatic Control*, (49) 9: 1465-1476, 2004.

[9] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag, 2001.

[10] R. L. Graham, M. Grötschel, and L. Lovász. *Handbook of Combinatorics*, The MIT Press/North-Holland, 1995.

[11] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control*, (48) 6: 988–1001, 2003.

[12] J. Komlós and M. Simonovits. Szemerédi's regularity lemma and its applications in graph theory, in *Combinatorics: Paul Erdős is Eighty*, vol. 2, Bolyai Society Mathematical Studies, 2, Budapest, 295–352, 1996.

[13] M. Mesbahi and F. Y. Hadaegh. Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching, *AIAA Journal of Guidance, Control, and Dynamics*, (24) 2: 369–377, 2001.

[14] M. Mesbahi. On a dynamic extension of the theory of graphs. *American Control Conference*, June 2002.

[15] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control*, (49) 9: 1520- 1533, 2004.

[16] S. V. Petrov. Graph grammars and graphodynamics problem. *Automation and Remote Control*, 10, 133-138, 1977.

[17] E. Prisner. *Graph Dynamics*. Longman House, England, 1995.

[18] G. Rozenberg (Editor). *Handbook of Graph Grammars and Computing by Graph Transformation*. World Scientific, Singapore, 1997.

[19] *http://planetquest.jpl.nasa.gov/TPF/tpf_index.html*

[20] H. Whitney. Congruent graphs and the connectivity of graphs, *American Journal of Mathematics*, 54, 150-168, 1932.