State synchronization in local-interaction networks is robust with respect to time delays

Herbert G. Tanner and Dimitrios K. Christodoulakis
Department of Mechanical Engineering
University of New Mexico
Albuquerque, NM 87131
Email: {tanner,dchristo}@unm.edu

***Abstract*— This is among the first results that suggest that multi-agent systems that interact through time-invariant, nearest-neighbor rules, can synchronize their states regardless of the size of communication delays. Network connectivity and consistency in the use of neighbor state information is all that is needed. The analysis in this paper is performed in discrete time and is based on the properties of non-negative matrices. Due to the fact that the state transition matrix is no more ergodic, the results of [1] cannot be applied. A crucial observation regarding the delayed dynamics allows the reduction of the model and the establishment of ergodicity for the system's state transition matrix.**

I. Introduction

It is generally true that communication delays degrade performance in networked control systems. In this paper we show that for a particular model of discrete time nearest-neighbor interaction, it is possible to intertwine control and communication in such a way that stability is robust with respect to delayed update of each neighbor state. In particular, if in the no-delay multi-agent system all agent states synchronize asymptotically, in the sense that all states converge to a common vector, the same will happen with the delayed model if these delays are bounded.

Interest in models of nearest-neighbor interaction (c.f. [2]–[4]) has been increasing recently, motivated by exciting links to biology, statistical mechanics [5], networking and computer science [6]. The stability of the Vicsek model [7] was investigated from a control's standpoint by Jadbabaie et. al, [1] in a seminal work that highlighted links between linear algebra, graph theory and cooperative control. Since then, kinematic [8] and dynamic [9]–[11] synchronization algorithms have been developed for multi-agent systems with local interaction rules. The stability of synchronization in large groups of coupled oscillators [12]–[14] has also been the focus of attention. In most of this work, information is generally assumed to be propagated instantly.

The effect of communication delays in networks of interconnected systems has been addressed in several contexts, one of the early ones being asynchronous distributed computation [15]. More recently, it has been investigated in internet-like coordination algorithms by Sandoval and Abdallah [16]. Liu and Passino [17] have considered the case of time delays and proved uniform ultimate boundedness of position errors in the swarm; velocity synchronization is pursued via gradient following. Saber and Murray [18] study the effect of delays on consensus protocols using Nyquist plots. These results suggest that communication delays have an adverse effect on stability. However, there is recent evidence that nearest neighbor interactions can be made robust to communication delays. Angeli et. al [19] extend the result of Moreau [20] including time delays. Using a set-valued Lyapunov function, they conclude that under certain compactness conditions, connectivity is sufficient to ensure stability. In ongoing work, Cao et. al [21] derive graph theoretic conditions for velocity syncrhonization under time delays, but both their model and underlying assumptions on the information agents can use, are slightly different. All these results seems to agree in principle with the one presented in this paper, but our approaches (developed in parallel) are different. In the model we are studying stability is established through the properties of ergodic matrices. Another difference is that in the problem addressed here the interconnection topology is fixed. Although our approach draws from the power of the results on nonnegative matrices, the treatment has to deviate significantly from that of [1], since due to the different state model, several matrix properties on which the proofs of the latter paper are built, are lost.

We look at a group of interconnected systems with single integrator dynamics in discrete time. Agents broadcast their state to their neighbors in turns, each having one time step to complete the transmission. Thus, only one agent transmits at a given time step and information cached by each agent about its neighbors can be outdated up to N steps. The systems update their state to the average of that of their neighbors, based on the *most recent information* cached. Under mild assumptions on the use of state updates, which ensure that neighbors use the same information regarding each other, we show that all systems synchronize their states to a common value, *regardless* of the time delay in the neighbor's state information. Our proof is based on the observation that some delayed information does not affect the state dynamics, and thus the state evolution can be described by a reduced model. Even though the original state matrix lacks several important properties, the one in the reduced model is shown to be ergodic and time invariant, from where the stability of the complete model follows. These stability results are supported by numerical simulations performed using the original (full) model.

In Section II we present the system at hand, we construct the discrete-time state space model and we identify the properties of the associated matrices. Section III presents our main result. In Section IV we illustrate the points of our proof with a numerical example. Section V provides simulation results showing the convergence of the system with delays for several configurations. Finally, Section VI concludes the paper with a summary of the results.

II. Problem Statement

The interactions between agents are represented by means of a time-invariant, undirected graph:

Definition II.1 (Interconnection graph) *The interconnection graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, is an undirected graph with:*

- *a set of vertices (nodes), $\mathcal{V} = \{n_1, \dots, n_N\}$, indexed by the agents in the group, and*
- *a set of edges, $\mathcal{E} = \{(n_i, n_j) \in \mathcal{V} \times \mathcal{V} \mid n_i \sim n_j\}$, which contains unordered pairs of nodes corresponding to interacting agents.*

The set of nodes that are adjacent to node i is denoted $\mathcal{N}_i$. The state of agent i is u_i and in the absence of any communication delays during agent interaction, it is updated in discrete time as follows [1]:

$$u_i(k+1) = \frac{1}{1+|\mathcal{N}_i|}\left(u_i(k) + \sum_{j\in\mathcal{N}_i} u_j(k)\right), \tag{1}$$

and collecting all states in a vector u, we can write the dynamics of the multi-agent system without delays as

$$u(k+1) = (\mathsf{I}+\mathsf{D})^{-1}(\mathsf{A}+\mathsf{I})u(k) \triangleq \mathsf{F}u(k), \tag{2}$$

where A and D are the adjacency and valency matrices of $\mathcal{G}$, respectively, [22]. To distinguish matrices from vectors we will use sans serif fonts for the former.

Note that the stability results obtained here are not dependent on the duration of the time step. Without loss of generality we can consider a numbering of nodes according to the order of transmission. Thus, every N steps, the transmission sequence is repeated. We will call the sequence of steps that starts with the transmission of agent 1 and ends with the transmission of agent N, a *communication cycle*.

To keep track of timed transmissions we will use a $0-1$ matrix S, with rows indexed by the agents and columns indexed by time steps: $s_{ij} = 1$ indicates that agent i transmits at cycle step j. As time progresses, the columns of S shift from left to right, with the rightmost column being recycled to the left. For example, for a group of four agents where agent 1 broadcasts first, then agent 2, then agent 3 and finally agent 4, at the time step k where agent 1 transmits again, and at time step $k+1$, $\mathsf{S}(k)$ and $\mathsf{S}(k+1)$ will have the form:

$$\mathsf{S}(k) = \begin{bmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0 \end{bmatrix}, \quad S(k+1) = \begin{bmatrix} 0&1&0&0\\ 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{bmatrix} \tag{3}$$

An agent i uses delayed neighbor states, and the earlier a particular neighbor j has broadcasted its state to i, the more delayed this state information is going to be when i uses it. In order to keep some consistency in state updates we will assume that agents are allowed to use their own current states *only after* they have broadcasted them to their neighbors.

Consider now augmenting the state vector $u(k)$ with components expressing previous states: $u(k-1), u(k-2), \dots, u(k-N+1)$. If we call

$$U(k) = (u(k)^T,\ u(k-1)^T, \dots, u(k-N+1)^T)^T,$$

then we can write the augmented system with delays as follows:

$$\begin{aligned} U(k+1) &= \begin{bmatrix} \multicolumn{6}{c}{(\mathsf{D}+\mathsf{I})^{-1}(\mathsf{I}+\mathsf{A})[S_i\otimes e_i^T]} \\ \mathsf{I} & 0 & \dots & \dots & 0 & 0 \\ 0 & \mathsf{I} & \dots & \dots & 0 & 0 \\ 0 & 0 & \mathsf{I} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & \dots & \mathsf{I} & 0 \end{bmatrix} U(k) \\ &\triangleq \mathsf{H}(k)U(k), \end{aligned} \tag{4}$$

where $[S_i \otimes e_i^T]$ is the matrix formed by S when its ith column, S_i, is replaced by the Kronecker product of itself with the transpose of the ith unit vector. For the example of four agents at time step k where $\mathsf{S}(k)$ is given by (3), the matrix $[S_i \otimes e_i^T]$ will have the form:

$$[S_i \otimes e_i^T] = \begin{bmatrix} 1\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,1\,0\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,1\,0 & 0\,0\,0\,0 \\ 0\,0\,0\,0 & 0\,0\,0\,1 & 0\,0\,0\,0 & 0\,0\,0\,0 \end{bmatrix}.$$

Note that despite the original (non-delay) system being time invariant, the delayed version is time varying, and H is changing between time steps due to the shift in the columns of S. In what follows, we will rewrite the first row of blocks of H as follows:

$$\mathsf{F}[S_i \otimes e_i^T] = \begin{bmatrix} \mathsf{f}_1 & \mathsf{f}_2 & \cdots & \mathsf{f}_N \end{bmatrix}.$$

Matrix H inherits some of the properties of F, namely: (i) it is row stochastic, and (ii) non-negative. However, it *does not have nonzero diagonal elements* and it is *not irreducible*. Without the latter two properties, the results of [1] do not apply and stability of (4) cannot be directly established.

The purpose of the following section will be to show that, despite the communication delays in state update, (4) synchronizes its state vector asymptotically, provided that the "nominal" (no-delay) system can do so.

III. Main Result

The system initializes at $k = 0$, without the agents having any knowledge of their neighbors' states. We have to allow one communication cycle to elapse for the agents to acquire local information. Our asymptotic analysis will therefore begin from $k = 1$. The following theorem summarizes the result of this paper:

Theorem III.1 (Stability with communication delays) *Consider a multi-agent system with a time invariant connected interconnection graph $\mathcal{G}$, and discrete time agent dynamics described by* (1)*. If only one agent is allowed to broadcast its state information to its neighbors at each time step k, and each agent can use its latest broadcasted own state in its update equations* (1)*, then in the resulting interconnected system with communication delays all agent states will converge to a single value.*

Proof: First note that the change in H from step to step is induced from the permutation of the first block of rows: $[\mathsf{f}_1 \ \cdots \mathsf{f}_N]$. Matrix H is periodic with a period of one communication cycle: $\mathsf{H}(t + N) = \mathsf{H}(t)$. Starting at every state, and sampling the trajectory at multiples of the communication cycle, the obtained sequence appears to be produced by a time invariant dicrete time system with the state transition matrix

$$\mathsf{M} \triangleq \mathsf{H}(N)\mathsf{H}(N-1)\cdots\mathsf{H}(1),$$

and since $\mathsf{H}(k)$ is row stochastic and non-negative for every k, M will also be (row) stochastic and non-negative. The state transition matrix of (4) can thus be written as

$$\Phi(kN, 1) = \mathsf{M}^k.$$

Let us now look into M closely. A direct calculation verifies that

$$\mathsf{M} = \begin{bmatrix} \mathsf{G}_1(N) & \mathsf{G}_2(N) & \cdots & \mathsf{G}_N(N) \\ \mathsf{G}_1(N-1) & \mathsf{G}_2(N-1) & \cdots & \mathsf{G}_N(N-1) \\ \vdots & \vdots & & \vdots \\ \mathsf{G}_1(1) & \mathsf{G}_2(1) & \cdots & \mathsf{G}_N(1) \end{bmatrix},$$

where $\mathsf{G}_i(1) = \mathsf{f}_i$, and the block elements of each column are given by the recursive formula:

$$\mathsf{G}_i(k) = \sum_{j=1} \mathsf{f}_j(k)\mathsf{G}_i(k-j), \tag{5}$$

where $\mathsf{G}_i(r)$ for $r < 1$ are read from the blocks of $\mathsf{H}(1)$ below $\mathsf{f}_i(1)$. Using (5), and given that

$$\begin{aligned} &\mathsf{f}_{\mathrm{mod}(i,N,1)}\big(\mathrm{mod}(j,N,1)\big) \\ &\qquad = \mathsf{f}_{\mathrm{mod}(i-1,N,1)}\big(\mathrm{mod}(j-1,N,1)\big), \end{aligned} \tag{6}$$

due to the permutation of S, where $\mathrm{mod}(\cdot, N, 1)$ is the modulo function with offset 1, we find that in all lower diagonal matrix blocks of M, including $\mathsf{G}_1(N), \mathsf{G}_2(N-1), \ldots, \mathsf{G}_N(1)$, we have $\mathsf{f}_i(1)$ appearing in the block column i as the last term in the sum $\sum_{j=1} \mathsf{f}_j(k)\mathsf{G}_i(k-j)$. All terms in this sum are right multiplied by $\mathsf{f}_i(1)$ and involve products of the form $\mathsf{f}_j(1)\mathsf{f}_i(1)$. Based on the special structure of $\mathsf{f}_i(1)$ every such multiplication will result (i) either in a zero matrix, if the node corresponding to the nonzero column of $\mathsf{f}_j(1)$ is not connected to the node of the nonzero column of $\mathsf{f}_i(1)$, meaning $f_{ji} = 0$,

$$\mathsf{f}_j\mathsf{f}_i = \begin{bmatrix} 0 & \ldots & 0 & f_{1j} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & f_{2j} & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \underbrace{0 \quad \ldots \quad 0}_{j-1 \text{ times}} & & & f_{Nj} & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} 0 & \ldots & 0 & f_{1i} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & f_{2i} & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \underbrace{0 \quad \ldots \quad 0}_{i-1 \text{ times}} & & & f_{Ni} & 0 & \ldots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \ldots & 0 & f_{1j}f_{ji} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & f_{2j}f_{ji} & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \underbrace{0 \quad \ldots \quad 0}_{i-1 \text{ times}} & & & f_{Nj}f_{ji} & 0 & \ldots & 0 \end{bmatrix},$$

or, (ii) if the nodes are connected, in a matrix having nonzero elements in every row of the *nonzero column of* $\mathsf{f}_i(1)$ for which $\mathsf{f}_j(1)$ has nonzero elements. In other words, if the corresponding nodes are connected, the product $\mathsf{f}_j(1)\mathsf{f}_i(1)$ will inherit nonzero elements from the nonzero column of $\mathsf{f}_j(1)$ and place them in the nonzero column of $\mathsf{f}_i(1)$.

With this said, it becomes clear that the remaining terms in each sum $\sum_{j=1} \mathsf{f}_j(k)\mathsf{G}_i(k-j)$ of block column i below the block diagonal, can only *add positive elements* in the nonzero column of $\mathsf{f}_i(1)$. In the G_i blocks above the diagonal of M, using (6) we can verify the existence of a term of the form $\mathsf{f}_i(1)^2$. This happens because the $\mathsf{f}_j(k)$ terms in $\sum_{j=1} \mathsf{f}_j(k)\mathsf{G}_i(k-j)$ are equated to the corresponding blocks $\mathsf{f}_r(1)$ via (6), with r taking values in $\{N-(k-2), \ldots, N, 1\}$. On the other hand, the diagonal block is located in block-row $N-(k-1)$, which means that some $\mathsf{f}_j(k)$ will be equal to $\mathsf{f}_i(1)$.

Since the diagonal element of the nonzero column of $\mathsf{f}_i(1)$ is always positive ($\mathsf{f}_i(1)$ inherits this property from F), $\mathsf{f}_i(1)^2$ will maintain the nonzero elements of $\mathsf{f}_i(1)$. Thus, for every $j = 1, \ldots, N$, we will have a nonzero column of M, m_i, with the following structure:

$$m_i = \Delta \begin{bmatrix} f_j(1) \\ \vdots \\ f_j(1) \end{bmatrix} + \lambda_i, \tag{7}$$

where Δ is a $N^2 \times N^2$ diagonal positive definite matrix, f_j is a column of F, and λ_i a nonnegative $N^2 \times 1$ vector.

The zero columns of M indicate that the associated states *do not contribute to the overall dynamics*. One can obtain a reduced model that describes the evolution of the system by removing those states from the state model. The reduced system, viewed over multiples of the communication cycle, has the form $\bar{U}(k+1) = \bar{\mathsf{M}}\bar{U}(k)$. In removing the zero columns from M, one column corresponding to a (possibly delayed) state of different agent is kept each time. Based on (7), $\bar{\mathsf{M}}$ will have the following form:

$$\bar{\mathsf{M}} = \Delta_1 \circ (\mathsf{FS}) + \Lambda,$$

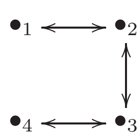
where Δ_1 is a positive $N \times N$ matrix, $\circ$ denotes the Hadamard product, Λ is a nonnegative $N \times N$ matrix and the product of F with the communication matrix S results in a permutation of the columns of F. With F being an irreducible matrix, since the underlying interaction graph of the nominal system is (strongly) connected, $\bar{\mathsf{M}}$ will be too: Λ will only insert additional edges in the underlying graph. With F having all diagonal elements nonzero, $\bar{\mathsf{M}}$ will be primitive too [23]. In addition, it is (row) stochastic, because the removal of zero columns from M does not affect the sum of the row elements. A primitive and stochastic matrix is ergodic [1] which means that by definition:

$$\lim_{k\to\infty} \bar{\mathsf{M}}^k = \mathbf{1}c^T,$$

which establishes the convergence of the reduced state vector to $\xi\mathbf{1}$, for some $\xi \in \mathbb{R}$. If current and (bounded) delayed states of agents converge to a common value, it is clear that all current states will converge. ■

IV. A Numerical Example

A group of four agents is interconnected as follows:



where a double headed arrow represents a bidirectional link (undirected edge). For this graph the matrix F in (2) is

$$\mathsf{F} = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 \\ 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix}$$

Assuming that the broadcasting sequence is $1, 2, 3, 4$ starting at step $k = 1$, the matrix H will have the form:

$$\mathsf{H}(1) = \begin{bmatrix} \mathsf{f}_1(1) & \mathsf{f}_2(1) & \mathsf{f}_3(1) & \mathsf{f}_4(1) \\ \mathsf{I} & 0 & 0 & 0 \\ 0 & \mathsf{I} & 0 & 0 \\ 0 & 0 & \mathsf{I} & 0 \end{bmatrix}$$

where

$$\mathsf{f}_1(1) = \begin{bmatrix} 1/2 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathsf{f}_2(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/3 \\ 0 & 0 & 0 & 1/2 \end{bmatrix},$$

$$\mathsf{f}_3(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}, \quad \mathsf{f}_4(1) = \begin{bmatrix} 0 & 1/2 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In the next time step, matrix H will evolve as follows:

$$\mathsf{H}(2) = \begin{bmatrix} \mathsf{f}_1(2) & \mathsf{f}_2(2) & \mathsf{f}_3(2) & \mathsf{f}_4(2) \\ \mathsf{I} & 0 & 0 & 0 \\ 0 & \mathsf{I} & 0 & 0 \\ 0 & 0 & \mathsf{I} & 0 \end{bmatrix}$$

where $\mathsf{f}_1(2) = \mathsf{f}_4(1)$, $\mathsf{f}_2(2) = \mathsf{f}_1(1)$, $\mathsf{f}_3(2) = \mathsf{f}_2(1)$, and $\mathsf{f}_4(2) = \mathsf{f}_3(1)$. In the following time steps we will have:

$$\mathsf{H}(3) = \begin{bmatrix} \mathsf{f}_3(1) & \mathsf{f}_4(1) & \mathsf{f}_1(1) & \mathsf{f}_2(1) \\ \mathsf{I} & 0 & 0 & 0 \\ 0 & \mathsf{I} & 0 & 0 \\ 0 & 0 & \mathsf{I} & 0 \end{bmatrix}$$

$$\mathsf{H}(4) = \begin{bmatrix} \mathsf{f}_2(1) & \mathsf{f}_3(1) & \mathsf{f}_4(1) & \mathsf{f}_1(1) \\ \mathsf{I} & 0 & 0 & 0 \\ 0 & \mathsf{I} & 0 & 0 \\ 0 & 0 & \mathsf{I} & 0 \end{bmatrix}.$$

Time step $k = 4$ signals the end of one communication cycle and matrix M will be:

$$\mathsf{M} \triangleq \mathsf{H}(4)\mathsf{H}(3)\mathsf{H}(2)\mathsf{H}(1).$$

In numerical form, it will read:

$$\mathsf{M} = \begin{bmatrix}
0.6667 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.1667 & 0 & 0 & 0.1667 & 0 & 0 \\
0.4815 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1111 & 0 & 0 & 0.2593 & 0 & 0 & 0.1481 & 0 & 0 \\
0.1667 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3333 & 0 & 0 & 0.3333 & 0 & 0 & 0.1667 & 0 & 0 \\
0.0833 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5000 & 0 & 0 & 0.3333 & 0 & 0 & 0.0833 & 0 & 0 \\
0.6667 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.1667 & 0 & 0 & 0.1667 & 0 & 0 \\
0.4815 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1111 & 0 & 0 & 0.2593 & 0 & 0 & 0.1481 & 0 & 0 \\
0.1481 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4444 & 0 & 0 & 0.2593 & 0 & 0 & 0.1481 & 0 & 0 \\
0.0556 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6667 & 0 & 0 & 0.2222 & 0 & 0 & 0.0556 & 0 & 0 \\
0.6667 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.1667 & 0 & 0 & 0.1667 & 0 & 0 \\
0.4444 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.4444 & 0 & 0 & 0.1111 & 0 & 0 \\
0.1111 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3333 & 0 & 0 & 0.4444 & 0 & 0 & 0.1111 & 0 & 0 \\
0.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5000 & 0 & 0 & 0.5000 & 0 & 0 & 0.0000 & 0 & 0 \\
0.5000 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.0000 & 0 & 0 & 0.5000 & 0 & 0 \\
0.3333 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0000 & 0 & 0 & 0.3333 & 0 & 0 & 0.3333 & 0 & 0 \\
0.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3333 & 0 & 0 & 0.3333 & 0 & 0 & 0.3333 & 0 & 0 \\
0.0000 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5000 & 0 & 0 & 0.5000 & 0 & 0 & 0.0000 & 0 & 0
\end{bmatrix}.$$

Notice the appearance of the zero columns, which also indicate that M is reducible. These columns will be maintained after multiplication from the left.

The matrix blocks of M displayed above can be calculated through the recursive formula (5), using $\mathsf{f}_i(j) = \mathsf{f}_{i-1}(j-1)$, as:

$$
\left[\begin{array}{c|c}
\begin{array}{r} f_1(4)\Big(f_1(3)\Big(f_1(2){+}I\Big){+}f_2(3){+}I\Big)f_1(1) \\ {+}f_2(4)\Big(f_1(2){+}I\Big)f_1(1) \\ {+}f_3(4)f_1(1){+}\mathbf{f_1(1)} \end{array} & \begin{array}{r} \mathbf{f_1(4)}\Big(f_1(3)\Big(f_1(2){+}I\Big) \\ {+}f_2(3){+}I\Big)\mathbf{f_2(1)} \\ {+}f_2(4)\Big(f_1(2){+}I\Big)f_2(1) \\ {+}f_3(4)f_2(1) \end{array} \\ \hline
\begin{array}{c} f_1(3)\Big(f_1(2){+}I\Big)f_1(1) \\ {+}f_2(3)f_1(1){+}\mathbf{f_1(1)} \end{array} & \begin{array}{c} f_1(3)\Big(f_1(2){+}I\Big)f_2(1) \\ {+}f_2(3)f_2(1){+}\mathbf{f_2(1)} \end{array} \\ \hline
f_1(2)f_1(1){+}\mathbf{f_1(1)} & f_1(2)f_2(1){+}\mathbf{f_2(1)} \\ \hline
\mathbf{f_1(1)} & \mathbf{f_2(1)}
\end{array}\right.
$$

$$
\left.\begin{array}{c|c}
\begin{array}{r} f_1(4)\Big(f_1(3)(f_1(2){+}I){+}f_2(3)\Big)f_3(1) \\ {+}\mathbf{f_2(4)}\Big(f_1(2){+}I\Big)\mathbf{f_3(1)}{+}f_3(4)f_3(1) \end{array} & \begin{array}{r} f_1(4)\Big(f_1(3)\Big(f_1(2){+}I\Big) \\ {+}f_2(3)\Big)f_3(1) \\ {+}f_2(4)f_1(2)f_4(1){+}\mathbf{f_1(2)f_4(1)} \end{array} \\ \hline
\mathbf{f_1(3)}\Big(f_1(2){+}I\Big)\mathbf{f_3(1)}{+}f_2(3)f_3(1) & f_1(3)f_1(2)f_4(1){+}\mathbf{f_1(2)f_4(1)} \\ \hline
f_1(2)f_3(1){+}\mathbf{f_3(1)} & \mathbf{f_1(2)f_4(1)} \\ \hline
\mathbf{f_3(1)} & \mathbf{f_4(1)}
\end{array}\right]
$$

Where we have printed in boldface and blue color the terms (or products of terms) that generate positive elements in the same locations as the matrix blocks of the last block row.

After 100×4 steps, the state transition matrix will be:

$$
M^{100} = \begin{bmatrix}
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0 \\
0.2857 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2857 & 0 & 0 & 0.2857 & 0 & 0 & 0.1429 & 0 & 0
\end{bmatrix},
$$

in which we can see that all rows have converged to the same (row) vector.

V. Simulations

In the examples we simulated, we used the dynamic equations given by (4) – not the reduced $\bar{M}$-dynamics. We tested three different interconnections of four agents with random initial conditions for the state, same for all examples:

$$
\begin{aligned}
U(0) = (&-30, 12, 5.6, 0.2, -9, 19, -23, 15, \\
&7, 3, -0.9, 1, 10, -15.6, 4, 5)^T
\end{aligned}
$$

The first interconnection is shown in (8),

$$
\begin{tikzcd}
\bullet_1 \arrow[r, leftrightarrow] & \bullet_2 \arrow[d, leftrightarrow] \\
\bullet_4 \arrow[r, leftrightarrow] & \bullet_3
\end{tikzcd}
\tag{8}
$$

in which the graph is minimally connected. The evolution of the (current) system states is depicted in Figure 1.

In the second example, the interconnection graph is complete:

$$
\begin{tikzcd}
\bullet_1 \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] \arrow[dr, leftrightarrow] & \bullet_2 \arrow[d, leftrightarrow] \arrow[dl, leftrightarrow] \\
\bullet_4 \arrow[r, leftrightarrow] & \bullet_3
\end{tikzcd}
$$

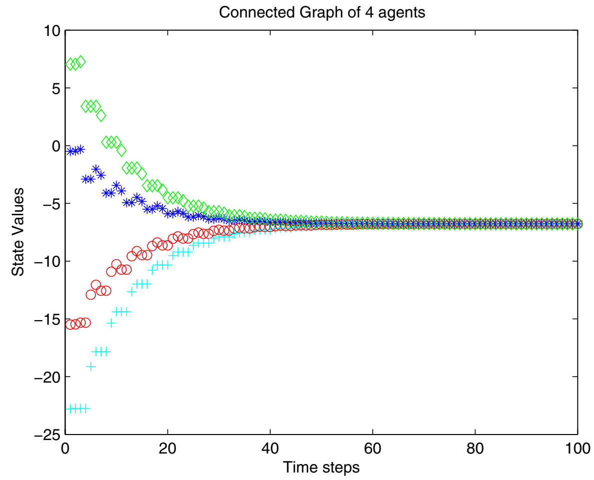


Fig. 1: Convergence in a minimally connected graph

The initial conditions are the same and the state trajectory is shown in Figure 2.

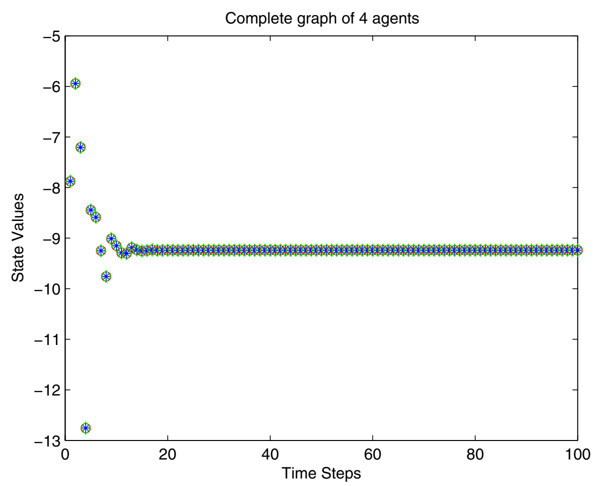


Fig. 2: Convergence in a complete graph.

One notes that in the case of the complete graph, all agent states get synchronized immediately, however, it takes some time until the synchronized value reaches a steady state. At every time step one cannot distinguish different markers in Figure 2, because they overlap. This behavior can be justified by a simple inspection of H.

In our last example, the interconnection graph is disconnected:

$$
\begin{tikzcd}
\bullet_1 \arrow[r, leftrightarrow] & \bullet_2 \\
\bullet_4 \arrow[r, leftrightarrow] & \bullet_3
\end{tikzcd}
$$

As expected, convergence takes place in each connected component, as shown in Figure 3. Since in this case, the connected components are complete graphs, all agent states within them synchronize from the beginning.

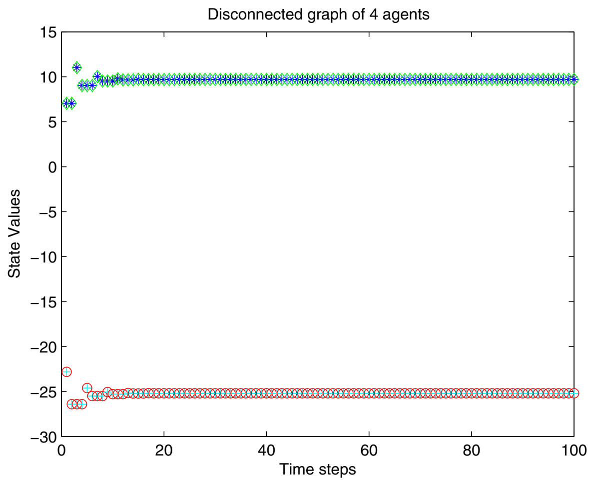


Fig. 3: Convergence in the connected components of a disconnected graph.

VI. Conclusions

In this paper we have shown that if an interconnected system of agents with time-invariant, nearest neighbor interaction laws of the form (1) is stable (in the sense that all agents states asymptotically synchronize), the introduction of communication delays in the dissemination of state information between neighboring agents does not disrupt the stability of the system. This was shown formally in discrete time, by exploiting properties of special classes of non-negative matrices and by observing that the system trajectories are not affected by parts of the full-blown delayed dynamics. The state matrix of this reduced system was then shown to be ergodic. To verify these findings, we tested the combined state communication/update algorithm in simulation and we found that the states in the (full-blown, delayed) dynamics asymptotically converged to the same, constant value. Extending these results to cases where the communication topology is time dependent will be among the objectives of our future work.

Acknowledgment

The authors would like to thank the Intelligent Systems & Robotics Center of the Sandia National Labs for their support under SURP grant # 342079 and specifically John Feddema, Ray Byrnes and Ken Groom.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Automat. Contr.*, vol. 48, pp. 988–1001, June 2003.

[2] Z. Lin, M. Broucke, and B. Francis, "Local control strategies for groups of mobile autonomous agents," *IEEE Transactions on Automatic Control*, vol. 49, no. 4, pp. 622–629, April 2004.

[3] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Stable flocking of mobile agents, Part I: Fixed topology," in *Proceedings of the IEEE Conference on Decision and Control*, Maui, Hawaii, December 9-12 2003, pp. 2010–2015.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

[5] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Science*, ser. Cambridge Nonlinear Science. Cambridge University Press, 2002.

[6] C. Reynolds, "Flocks, birds, and schools: a distributed behavioral model," *Computer Graphics*, vol. 21, pp. 25–34, 1987.

[7] T. Vicsek, A. Czirok, E. B. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Physical Review Letters*, vol. 75, pp. 1226–1229, 1995.

[8] Y. Liu, K. M. Passino, and M. M. Polycarpou, "Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology," *IEEE Trans. Automat. Contr.*, vol. 48, pp. 76–95, Jan. 2003.

[9] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *Automatica*, July 2003, (submitted).

[10] ——, "Flocking in teams of nonholonomic agents," in *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences, S. Morse, N. Leonard, and V. Kumar, Eds. Springer, 2004, pp. 229–239.

[11] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE 40th Conf. Decision and Control*, Orlando, Florida, Dec. 2001, pp. 2968–2973.

[12] S. H. Strogatz, "From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators," *Physica D*, vol. 143, pp. 1–20, 2000.

[13] A. Jadbabaie, N. Motee, and M. Barahona, "On the stability of the kuramoto model of coupled nonlinear oscillators," in *Proceedings of the IEEE American Control Conference*, Boston, MA, July 2004, pp. 4296–4301.

[14] R. Sepulchre, D. Paley, and N. Leonard, "Collective motion and oscillator synchronization," in *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences, S. Morse, N. Leonard, and V. Kumar, Eds. Springer, 2004, pp. 189–206.

[15] N. J. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, September 1986.

[16] R. Sandoval-Rodriguez, C. T. Abdallah, and P. F. Hokayem, "Internet-like protocols for the control and coordination of multiple agents with time delay," in *IEEE International Symposium on Intelligent Control*, Houston, TX, Oct. 2003.

[17] Y. Liu and K. M. Passino, "Cohesive behaviors of multiagent systems with information flow constraints," *IEEE Transactions on Automatic Control*, 2004, submitted.

[18] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multi-vehicle systems," in *Proc. IEEE 41st Conf. Decision and Control*, Las Vegas, Nevada, Dec. 2002, pp. 2965–2971.

[19] D. Angeli and P.-A. Bliman, "Stability of leaderless multi-agent systems. extension of a result by Moreau," math.OC/0411338, (http://arxiv.org/list/math.oc/0411).

[20] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.

[21] M. Cao, B. Anderson, and S. Morse, "Unpublished correspondence," 2005.

[22] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York, NY: Springer-Verlag, 2001.

[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, UK: Cambridge University Press, 1999.