Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Agreement via the Edge Laplacian

Daniel Zelazo, Amirreza Rahmani, and Mehran Mesbahi†

***Abstract*— This work explores the properties of the edge variant of the graph Laplacian in the context of the *edge agreement* problem. We show that the edge Laplacian, and its corresponding agreement protocol, provide a useful perspective on the well-known *node agreement*, or the consensus problem. Specifically, the dynamics induced by the edge Laplacian facilitates a better understanding of the role played by certain subgraphs, e.g., cycles and spanning trees, in the original agreement problem. We also point out a reduced order modeling of the edge agreement as parameterized by the spanning trees of the underlying graph.**

I. Introduction

Distributed dynamic systems are collections of dynamical units that interact over an information exchange network. Such systems are ubiquitous in diverse areas of science and engineering. Examples include physiological systems and gene networks [1], large scale energy systems, and multiple space, air, and land vehicles [2], [3], [4], [5], [6]. There is an active research effort underway in the control and dynamical systems community to formalize these systems and lay out a foundation for their analysis and synthesis [7], [8], [9]. As a result of this effort, a distinct area of research that lies at the intersection of systems theory and graph theory has emerged. A basic yet fundamental class of problems that lies at this intersection relates to the Laplacian dynamics, also known as the agreement or consensus protocol [10], [11], [12], [13].

The Laplacian dynamics is the process via which a group of first order units, through local interactions, reach a common value of interest. In this paper we explore the properties of the edge variant of this well-known protocol and refer to it as the *edge agreement* problem. We show that the edge Laplacian, and its corresponding edge agreement protocol, provide a new perspective on the Laplacian dynamics. Specifically, edge agreement facilitates a better understanding of the role played by certain subgraphs, e.g., cycles and spanning trees, in the node agreement problem. We also point out a reduced order modeling for the edge agreement in terms of the spanning trees of the original graph.

II. Preliminaries and Notations

We first provide a few notions that will be employed throughout the paper for studying node and edge agreement problems.

†This work was supported by the National Science Foundation grants CMS-0301753 and ECS-0501606. The authors are with the Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195-2400. Emails: {*dlz* + *rahmani*}*@u.washington.edu* and *mesbahi@aa.washington.edu*.

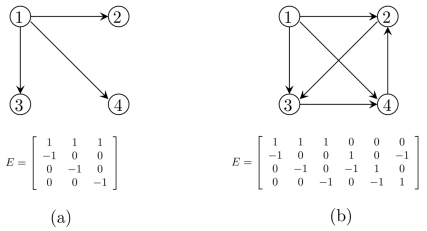

Fig. 1. Incidence matrices for two simple graphs

A. Graphs and Their Algebraic Representation

An undirected (simple) graph $\mathcal{G}$ is specified by a vertex set $\mathcal{V}$ and an edge set $\mathcal{E}$ whose elements characterize the incidence relation between distinct pairs of $\mathcal{V}$. Two vertices i and j are called *adjacent* (or neighbors) when $\{i, j\} \in \mathcal{E}$; we denote this by writing $i \sim j$. The cardinalities of the vertex and edge sets of $\mathcal{G}$ will be denoted by $|\mathcal{G}|$ and $\|\mathcal{G}\|$, respectively. A *subgraph* of a graph $\mathcal{G}$ is a graph whose vertex and edge sets are subsets of those of $\mathcal{G}$. An *orientation* of an undirected graph $\mathcal{G}$ is the assignment of directions to its edges, i.e., an edge e_k is an ordered pair (i, j) such that i and j are, respectively, the initial and the terminal nodes of e_k.

Graphs admit a set of convenient matrix representations. For example, the $|\mathcal{G}| \times \|\mathcal{G}\|$ incidence matrix $E(\mathcal{G})$ for an oriented graph $\mathcal{G}$ is a $\{0, \pm 1\}$-matrix with rows and columns indexed by vertices and edges of $\mathcal{G}$, respectively, such that

$$[E(\mathcal{G})]_{ik} = \begin{cases} +1 & \text{if } i \text{ is the initial node of edge } e_k \\ -1 & \text{if } i \text{ is the terminal node of edge } e_k \\ 0 & \text{otherwise} \end{cases}$$

Figure 1 depicts an example of two oriented graphs and their respective incidence matrices. From the definition of the incidence matrix it follows that the null space of its transpose, $\mathcal{N}(E(\mathcal{G})^T)$, contains **span** $\{\mathbf{1}\}$, where $\mathbf{1}$ is the vector with all entries equal to one with appropriate dimensions. The rank of the incidence matrix depends only on $|\mathcal{G}|$ and the number of its connected components [14].

Theorem 2.1: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with c connected components. Then **rank** $E(\mathcal{G}) = |\mathcal{G}| - c$.

The degree of a vertex is the cardinality of the set of vertices adjacent to it. A graph is complete if all possible pairs of vertices are adjacent, or equivalently, if the degree of all vertices is $|\,\mathcal{G}\,| - 1$. A sequence of $r + 1$ distinct and consecutively adjacent vertices, starting from vertex i and ending at vertex j, is called a path of length r (form i to j); when $i = j$, we call this path a *cycle*. We call a graph *connected* if there exists a path between any pair of vertices.

A connected graph without cycles is referred to as a *tree*. Equivalently, a tree is a connected graph on $|\mathcal{G}|$ vertices with $|\mathcal{G}|-1$ edges. Figure 1(a) shows an example of a tree, while Figure 1(b) is a connected graph containing cycles.

The Laplacian of $\mathcal{G}$, $L(\mathcal{G}) := E(\mathcal{G})\,E(\mathcal{G})^T$, is a rank deficient positive semi-definite matrix. The real spectrum of the Laplacian can thereby be ordered as

$$0 = \lambda_1(L(\mathcal{G})) \leq \lambda_2(L(\mathcal{G})) \leq \ldots \leq \lambda_{|\mathcal{G}|}(L(\mathcal{G})).$$

A direct consequence of Theorem 2.1 is that the multiplicity of the zero eigenvalue of the graph Laplacian is equal to the number of connected components of the graph [14]. Moreover, the second smallest eigenvalue of $L(\mathcal{G})$, $\lambda_2(L(\mathcal{G}))$, also known as algebraic connectivity, turns out to be a judicious measure of graph connectivity [17].

Since the dependency of the subsequent derivations on the underlying graph is implicit, we will not include "$\mathcal{G}$" in the notation used for matrices associated with the graph.

B. Agreement Dynamics

The agreement protocol is one of the basic yet fundamental problems in multi-agent coordination. It is built upon a general setup consisting of a group of N dynamic units, each connected to a fixed number of other units in the ensemble. Labeling these units as 1 through N, the interconnection between the dynamic units can be represented by a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with $\mathcal{V} = \{1,\ldots,N\}$ and $\mathcal{E}$ denoting the set of pairwise inter-unit couplings. Each unit is assumed to have a simple first order dynamics of the form

$$\dot{x}_i(t) = z_i(t), \quad i = 1,\ldots,N. \tag{1}$$

The interaction or coupling between units' dynamics is realized through the control input $z_i(t)$ in (1), assumed to be the sum of the differences between states of the unit and its neighbors, i.e.,

$$z_i(t) = \sum_{i\sim j}(x_j(t) - x_i(t)). \tag{2}$$

Expressing the dynamic evolution of the resulting system in a compact matrix form with $\mathbf{x}_n(t) = [\,x_1(t),\ldots,x_N(t)\,]^T$, one has

$$\dot{\mathbf{x}}_n(t) = -L_n\,\mathbf{x}_n(t), \tag{3}$$

where $\mathbf{x}_n$ denotes the collection of node states[1] and $L_n = L(\mathcal{G})$.

The agreement set $\mathcal{A} \subseteq \mathbb{R}^N$ is the subspace $\mathbf{span}\,\{\mathbf{1}\}$. Let us define $\delta_n(t)$ as the projection of states $\mathbf{x}_n(t)$ onto the subspace orthogonal to the agreement subspace. This subspace is denoted by $\mathbf{1}^\perp$; in [13] it is referred to as the *disagreement* subspace. Thus

$$\delta_n(t) = \mathbf{x}_n(t) - \alpha\,\mathbf{1},$$

where $\alpha = (1/N)\sum_i (x_n)_i$.

Proposition 1 ([13]): The Laplacian dynamics (3) converges to the agreement subspace from an arbitrary initial condition if and only if the underlying graph is connected.

[1]The subscript n signifies that the graph Laplacian is the system matrix for the "node" agreement.

III. Edge Laplacian and its Properties

In this section we introduce an edge variant of the graph Laplacian for a group of interconnected units. Furthermore, we explore some of the connections between the *edge Laplacian* and the well-studied graph Laplacian. The dynamic relevance of edge Laplacian in the edge agreement will be considered in §IV.

The edge Laplacian of an arbitrary oriented graph $\mathcal{G}$ is defined as

$$L_e := E^T\,E. \tag{4}$$

The edge Laplacian is a real $\|\mathcal{G}\| \times \|\mathcal{G}\|$ symmetric matrix. Two immediate linear algebraic properties of L_e are given below:

- the non-zero eigenvalues of L_e are equal to the non-zero eigenvalues of L_n, and
- The non-zero eigenvalues of L_e and L_n are equal to the square of the non-zero singular values of $E(\mathcal{G})$.

From these properties and Theorem 2.1 it follows that the rank of the edge Laplacian is also related to the number of connected components in the graph. The edge Laplacian can be permuted into a block diagonal matrix where each block represents a connected component.

Lemma 3.1: Consider the graph $\mathcal{G}$ with p connected components, $\mathcal{G}_i$, and associated incidence matrices, E_i, and let $E = \left[\begin{array}{ccc} E_1 & \ldots & E_p \end{array}\right]$. Then the edge Laplacian for $\mathcal{G}$ has the block diagonal form

$$L_e = \left[\begin{array}{ccc} E_1^T E_1 & & \\ & \ddots & \\ & & E_p^T E_p \end{array}\right]. \tag{5}$$

Proof: Denote each column (corresponding to an edge) of the incidence matrix by the vector e_i. Then

$$e_i^T e_j = \begin{cases} \pm 1 & \text{if } e_i \text{ and } e_j \text{ share a single node} \\ 2 & \text{if } i = j \\ 0 & \text{if } e_i \text{ and } e_j \text{ share no nodes} \end{cases}$$

By definition two connected components of a graph do not share a node; hence the corresponding entry in the edge Laplacian is zero.

■

Proposition 2: Adding an edge to the graph $\mathcal{G}$ increases the sum of the eigenvalues of the corresponding graph and edge Laplacians by 2.

Proof: One way to prove this proposition is via the graph Laplacian. The sum of eigenvalues of the Laplacian is the sum of its diagonal entries- that in turn- is twice the number of edges of $\mathcal{G}$. Adding an edge to $\mathcal{G}$ therefore adds 2 to this sum. Yet another way to reach the same conclusion is via the edge Laplacian. Adding an edge to $\mathcal{G}$ adds a column to the incidence matrix of $\mathcal{G}$, i.e., $E(\hat{\mathcal{G}}) = [\,E(\mathcal{G}) \quad e\,]$ and will augment row and column borders to the original edge Laplacian as,

$$L_e(\hat{\mathcal{G}}) = \left[\begin{array}{cc} L_e(\mathcal{G}) & E^T e \\ e^T E & 2 \end{array}\right]. \tag{6}$$

As a result,

$$\begin{aligned} \textbf{trace } L_e(\hat{\mathcal{G}}) &= \sum_i \lambda_i(L_e(\hat{\mathcal{G}})) \\ &= \textbf{trace } L_e(\mathcal{G}) + 2 \end{aligned}$$

■

Recall that the graph Laplacian is always rank-deficient. In the edge case, we find that the rank of the edge Laplacian depends only on the number of connected components, as shown in Theorem 2.1. Furthermore, the null space of the edge Laplacian depends on the number of cycles in the graph. Let us elaborate on this last statement by presenting a few definitions and theorems.

Definition 3.1: Given an incidence matrix E for a directed graph, a *signed path vector* is a vector $z \in \mathbb{R}^{\|\mathcal{G}\|}$ corresponding to a path such that

$$z_i = \begin{cases} 1 & \text{if edge } i \text{ is traversed positively} \\ -1 & \text{if edge } i \text{ is traversed negatively} \\ 0 & \text{if edge } i \text{ is not used in the path} \end{cases}$$

Lemma 3.2: Given a path with distinct initial and terminal nodes described by a signed path vector z in a graph $\mathcal{G}$, the vector $y = Ez$ has the following structure:

$$y_i = \begin{cases} 1 & \text{if node } i \text{ is the initial node of path } z \\ -1 & \text{if node } i \text{ is the final node of path } z \\ 0 & \text{otherwise} \end{cases}$$

Proof: We can rewrite Ez as $E\,\textbf{diag}(z)\,\mathbf{1}$. The ij-th entry of the matrix $E\,\textbf{diag}(z)$ will be -1 if edge j is used by the path to leave node i, $+1$ if edge j is used by the path to enter node i, and zero otherwise. If node i is an intermediate node in the path (neither the initial nor the terminal node), then the path must enter and leave the node an equal number of times, resulting in the i-th row-sum of $E\,\textbf{diag}(z)$ to be zero. On the other hand, if node i is the initial node, the path must eventually leave the node without ever returning to it, resulting in the i-th row-sum to be equal to 1. Similarly, if i is the terminal node, the path must eventually enter the node without ever leaving it, resulting in the i-th row-sum to be equal to -1.

■

Theorem 3.3: Given a connected directed graph $\mathcal{G}$ with incidence matrix E, the null space of E is spanned by all the linearly independent signed path vectors corresponding to the cycles in E.

Proof: For any node used in a cycle path, the path must enter and exit that node an equal number of times. Using the same argument as in the proof of Lemma 3.2, it follows that $Ez = \mathbf{0}$ when z is the signed path vector for a cycle.

■

Theorem 3.3 is an example of the intricate relationship between the graphical and algebraic properties of a graph.

Theorem 3.4: Let L_e and E be, respectively, the edge Laplacian and the incidence matrix of an oriented graph. Then

$$\mathcal{N}(L_e) = \mathcal{N}(E). \tag{7}$$

Proof: Let $x \in \mathcal{N}(E)$; then $L_e x = E^T E x = \mathbf{0}$ and it follows that $\mathcal{N}(E) \subseteq \mathcal{N}(L_e)$. On the other hand when $x \in \mathcal{N}(L_e)$, $E^T E x = \mathbf{0}$ and $x^T E^T E x = \|Ex\|^2 = 0$. Thus $x \in \mathcal{N}(E)$.

■

IV. The Edge Agreement Problem

The agreement problem has been studied extensively from the perspective of the node states in the systems and control literature. In many applications node agreement is the natural venue to study the dynamic states of a group of interconnected first order agents. This problem has a natural edge interpretation that we will explore in this section.

We consider the system states to be on the edges rather than on the nodes. The node-interpretation can be used to gain insight on the edge version by noting that the edge state represents the difference between the two nodes incident to an edge. In this avenue, let

$$\mathbf{x}_e(t) = E^T \mathbf{x}_n(t) \tag{8}$$

where $\mathbf{x}_e(t) \in \mathbb{R}^{\|\mathcal{G}\|}$ represents the relative, or edge state. Differentiating (8) and substituting in (3) leads to

$$\begin{aligned} \dot{\mathbf{x}}_e(t) &= E^T \dot{x}_n(t) \\ &= -L_e \mathbf{x}_e(t). \end{aligned} \tag{9}$$

We now examine the agreement protocol in the edge setting (9). If we use the node-to-edge transformation above, we see that agreement is equivalent to having $\mathbf{x}_e(t) = \mathbf{0}$. As a result, in the edge setting the disagreement, denoted by $\delta_e(t)$, rather than being the distance to a subspace as in the node case- is the distance to the origin, i.e.,

$$\|\delta_e(t)\|_2 = \|\mathbf{x}_e(t)\|_2 \leq \|E\|_2 \|\delta_n(t)\|_2 \tag{10}$$

From Proposition 1, the node dynamics converges to the agreement subspace, which implies that the edge dynamics converges to the origin. Hence, we can view the agreement protocol as driving the edge states to the origin. Analogous to the node agreement, in the edge agreement setting, the evolution of an edge state depends on its current state and the states of its adjacent edges, i.e., those that share a node with it.

A. Role of Cycles in Edge Agreement

Cycles of the graph play an important role in both the node and edge version of the corresponding Laplacian dynamics. Recall that the null space of the edge Laplacian characterizes the cycle space of the graph. In the meantime, in the Laplacian dynamics, agreement is reached when the underlying state trajectory converges to the null-space of L_n. The same observation is valid in the dynamics dictated by the edge Laplacian, that is, when $\mathbf{x}_e(t) \in \mathcal{N}(L_e)$, agreement has been reached.

To simplify our presentation we will assume that all graphs under consideration are connected and hence contain a spanning tree. The edges that are not in the given spanning

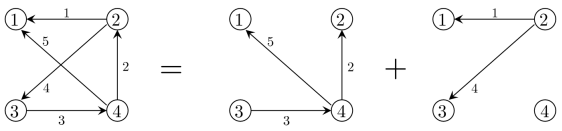

Fig. 2. A graph can be represented (not necessarily in a unique way) as a tree and edges that complete its cycles

tree must complete the cycles in the graph. Using an appropriate permutation of the edge ordering, we can express the incidence matrix as

$$E = \left[\begin{array}{cc} E_t & E_c \end{array} \right] \tag{11}$$

where E_t represents a given spanning tree and E_c represents the remaining edges not in the tree (the cycle edges). Note that in general, E_c does not represent a connected graph.

It is useful to express the node and edge Laplacians in terms of the block representation of the incidence matrix as

$$\begin{aligned} L_n &= \left[\begin{array}{cc} E_t & E_c \end{array} \right] \left[\begin{array}{cc} E_t & E_c \end{array} \right]^T \\ &= E_t E_t^T + E_c E_c^T \end{aligned} \tag{12}$$

and

$$\begin{aligned} L_e &= \left[\begin{array}{cc} E_t & E_c \end{array} \right]^T \left[\begin{array}{cc} E_t & E_c \end{array} \right] \\ &= \left[\begin{array}{cc} E_t^T E_t & E_t^T E_c \\ E_c^T E_t & E_c^T E_c \end{array} \right]. \end{aligned} \tag{13}$$

Our first observation is that the node agreement has two components: one corresponding to the "tree dynamics," and the other to the "cycle dynamics." This is represented graphically in Figure 2.

An important property of both the node and edge agreement is the rate at which the system converges to the agreement subspace. As the smallest non-zero eigenvalue of the graph Laplacian, i.e, $\lambda_2(L_n)$, dictates this rate, it becomes of interest to determine efficient means of designing or modifying a network to increase $\lambda_2(L_n)$. The form of (12) suggests a transparent way by which the cycles of the graph contribute to the overall convergence rate of the system.

Theorem 4.1: Consider the respective ordered eigenvalues of the graph Laplacian L_n, as defined in (12), and the corresponding tree Laplacian $E_t\, E_t^T$. Then for $i = 1, \ldots, N$, one has

$$\lambda_i(L_n) \geq \lambda_i(E_t\, E_t^T).$$

Proof: By construction, L_n, $E_t\, E_t^T$, and $E_c\, E_c^T$ are positive semi-definite matrices. From (12) one can see that the matrix difference $L_n - E_t\, E_t^T$ is positive semi-definite. The proof now follows from Corollary 7.7.4c of [15]. More specifically, we note that

$$\lambda_2(L_n) \geq \lambda_2(E_t\, E_t^T).$$

■

This result points to the importance of cycles as they pertain to rapidity of convergence to the agreement subspace. It also suggests that convergence to the steady-state value can be studied using a simpler graph (i.e., a graph with less edges) if the rate of convergence is not of prime concern. This observation has implications for computational complexity and controller/estimator designs, which will be explored in our future work.

The edge Laplacian also provides a matrix representation for substantiating on Theorem 4.1 by using the interlacing eigenvalues theorem for bordered matrices [15].

Theorem 4.2: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a tree represented by the edge Laplacian L_e with an ordered spectrum $\lambda_1 \leq \cdots \leq \lambda_{\mathcal{G}}$. Adding an edge from the complement of $\mathcal{E}$ to the tree leads to a new graph $\hat{\mathcal{G}}$ with edge Laplacian $\hat{L}_e$ and an ordered spectrum $\hat{\lambda}_1 \leq \cdots \leq \hat{\lambda}_{|\mathcal{G}|+1}$, satisfying the following inequalities:

$$0 = \widehat{\lambda}_1 \leq \lambda_1 \leq \widehat{\lambda}_2 \leq \lambda_2 \leq \cdots \leq \lambda_{|\mathcal{G}|} \leq \widehat{\lambda}_{|\mathcal{G}|+1}. \tag{14}$$

Proof: First, recall from Theorem 3.3 that cycles in a graph correspond to the null space of the edge Laplacian. Adding an edge from the complement of $\mathcal{E}$ to the graph (which is assumed to be a tree) must therefore create a cycle. The remainder of the proof follows from the application of the Courant-Fischer Theorem [15] in the context of the interlacing property of the eigenvalues of L_e and $\widehat{L}_e$.

■

Theorems 4.1 and 4.2 provide insights as to when it may be beneficial to add an edge for the purpose of improving the convergence rate of the node agreement. One immediate observation is that under certain circumstances, adding an edge does not improve the convergence rate.

Proposition 3: Given a graph $\mathcal{G}$ with incidence matrix E and edge Laplacian L_e, if the smallest non-zero eigenvalue has multiplicity $m > 1$, then adding up to m new edges will *not* increase the smallest non-zero eigenvalue of the new graph.

Proof: Let us consider the case of $m = 2$, as the proof for general m is analogous. Assume, without loss of generality, that $\mathcal{G}$ is a tree. By assumption, we have that $\lambda_1(L_e) = \lambda_2(L_e) \neq 0$. Using the results of Theorem 4.2, we have that

$$0 = \widehat{\lambda}_1 \leq \lambda_1 \leq \widehat{\lambda}_2 \leq \lambda_2$$

where $\widehat{\lambda}_i$'s are the corresponding edge Laplacian eigenvalues of the augmented graph. However, since $\lambda_1 = \lambda_2$, we must have

$$\lambda_1 = \widehat{\lambda}_2 = \lambda_2.$$

■

As a simple example, consider the graph in Figure 1(a). The smallest eigenvalue for the edge Laplacian of this graph is one with multiplicity two. Hence adding a single edge to this graph will not improve the smallest eigenvalue of its edge Laplacian. This observation also suggests that certain spanning trees will naturally lead to faster convergence rates. As an example, a star graph (shown in Figure 1(a)) will have a greater smallest eigenvalue than a path graph.

B. Reduced Order Edge Agreement

In the previous section we showed a connection between the cycles of a graph and the algebraic structure of its corresponding edge Laplacian. These results can be used to derive a reduced order representation of the edge agreement over a graph in terms of the corresponding dynamics on its spanning tree subgraph. We define the edge state vector to be

$$\mathbf{x}_e(t) = \left[\begin{array}{cc} \mathbf{x}_t^T(t) & \mathbf{x}_c^T(t) \end{array} \right]^T,$$

where $\mathbf{x}_t(t)$ is the edge state of the tree subgraph E_t and $\mathbf{x}_c(t)$ are the remaining edge states (the cycle states). Note that $\mathbf{x}_t(t) \in \mathbb{R}^{|\mathcal{G}|-1}$ and $\mathbf{x}_c(t) \in \mathbb{R}^{\|\mathcal{G}\|-|\mathcal{G}|+1}$. Reaching agreement corresponds to having $\mathbf{x}_t(t) = \mathbf{0}_{|\mathcal{G}|-1}$ and $\mathbf{x}_c(t) = \mathbf{0}_{\|\mathcal{G}\|-|\mathcal{G}|+1}$.

Theorem 4.3: Consider a graph $\mathcal{G}$ with cycles, and a tree subgraph $\mathcal{G}_t$, with corresponding edge Laplacians $L_e(\mathcal{G}) = \left[\begin{array}{cc} E_t & E_c \end{array} \right]^T \left[\begin{array}{cc} E_t & E_c \end{array} \right]$ and $L_e(\mathcal{G}_t) = E_t^T E_t$, respectively. Then there exists a matrix R such that

$$L_e(\mathcal{G}) = R^T L_e(\mathcal{G}_t) R. \tag{15}$$

Proof: The graph $\mathcal{G}$ has cycles which implies that the columns of E_c are linearly dependent on the columns of E_t. This dependency can be expressed in terms of the existence of a matrix T such that

$$E_t T = E_c. \tag{16}$$

Since E_t has full column rank, its pseudo-inverse exists and we have

$$T = (E_t^T E_t)^{-1} E_t^T E_c. \tag{17}$$

Therefore, the incidence matrix for $\mathcal{G}$ can be written as

$$E = \left[\begin{array}{cc} E_t & E_t T \end{array} \right]. \tag{18}$$

We can calculate the edge Laplacian for $\mathcal{G}$ in terms of the matrices E_t and T as

$$\begin{aligned} L_e(\mathcal{G}) &= \left[\begin{array}{cc} E_t^T E_t & E_t^T E_t T \\ T^T E_t^T E_t & T^T E_t^T E_t T \end{array} \right] \\ &= \left[\begin{array}{c} I_{\|\mathcal{G}_t\|} \\ T^T \end{array} \right] L_e(\mathcal{G}_t) \left[\begin{array}{cc} I_{\|\mathcal{G}_t\|} & T \end{array} \right]. \end{aligned}$$

The matrix R can now be defined as

$$R = \left[\begin{array}{cc} I_{\|\mathcal{G}_t\|} & T \end{array} \right]. \tag{19}$$

■

Theorem 4.3 makes it transparent that cycles are not necessary for convergence in the edge Laplacian dynamics. In fact, all cycle states can be reconstructed from the tree states through the linear relationship derived in (17). This leads us to the following result.

Theorem 4.4: Consider a graph $\mathcal{G}$ with incidence matrix $E = \left[\begin{array}{cc} E_t & E_c \end{array} \right]$ where E_t is the incidence matrix for a tree subgraph $\mathcal{G}_t$ and E_c represents the incidence matrix for the cycle edges. Partition the edge state vector as $\mathbf{x}_e(t) = \left[\begin{array}{cc} \mathbf{x}_t^T(t) & \mathbf{x}_c^T(t) \end{array} \right]^T$. Then the system described by

$$\dot{\mathbf{x}}_e(t) = -L_e(\mathcal{G}) \mathbf{x}_e(t) \tag{20}$$

is equivalent to the system described by

$$R^T \dot{\mathbf{x}}_t(t) = -R^T L_e(\mathcal{G}_t) R R^T \mathbf{x}_t(t) \tag{21}$$

where R is as defined in (19).

Furthermore, the reduced order system described by

$$\dot{\mathbf{x}}_t(t) = -L_e(\mathcal{G}_t) R R^T \mathbf{x}_t(t) \tag{22}$$

captures the behavior of the system described in (20). In fact the cycle edge states can be reconstructed by using the matrix T as defined in (17) via

$$\mathbf{x}_c(t) = T^T \mathbf{x}_t(t). \tag{23}$$

Proof: Using (12) and (18), the Laplacian dynamics can be written as

$$\dot{\mathbf{x}}_n(t) = -(E_t E_t^T + E_t T T^T E_t^T) \mathbf{x}_n(t). \tag{24}$$

The edge dynamics can then be derived by recalling that

$$\begin{aligned} \mathbf{x}_e(t) &= E^T \mathbf{x}_n(t) \\ &= \left[\begin{array}{c} E_t^T \\ T^T E_t^T \end{array} \right] \mathbf{x}_n(t) \\ &= R^T E_t^T \mathbf{x}_n(t). \end{aligned}$$

Left-multiplying (24) by $R^T E_t^T$ leads to

$$R^T \dot{\mathbf{x}}_t(t) = -R^T (E_t^T E_t + E_t^T E_t T T^T) \mathbf{x}_t(t)$$

which is the desired result (21). The reduced order representation follows directly from the structure of the matrix R.

■

V. Simulations

In this section we show a set of simulation results for some of the properties discussed in previous sections. To explore the role of cycles and adding extra edges to a tree, we simulated the edge agreement problem for the graphs shown in Figure 3. The graph of Figure 3(a) is a tree. It can be easily verified that adding an edge, as in Figure 3(b), creates a cycle in the graph but the smallest non-zero eigenvalue of the Laplacian remains unchanged. If a second edge is added, as in Figure 3(c), then the smallest eigenvalue increases.

Figure 4 depicts the trajectories of the edge states for graphs shown in Figure 3. We note that the slowest mode of convergence for graphs in Figure 3(a) and Figure 3(b) are the same, while the graph with two extra edges reaches agreement much faster. This shows that in certain network configurations (here the tree was a star graph), adding only

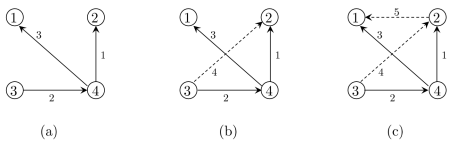

Fig. 3. (a) a tree star graph, (b) tree with an extra edge, and (c) tree with two extra edges

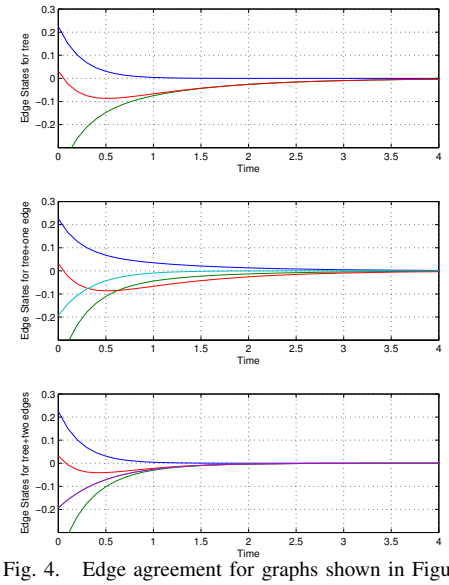

Fig. 4. Edge agreement for graphs shown in Figure 3

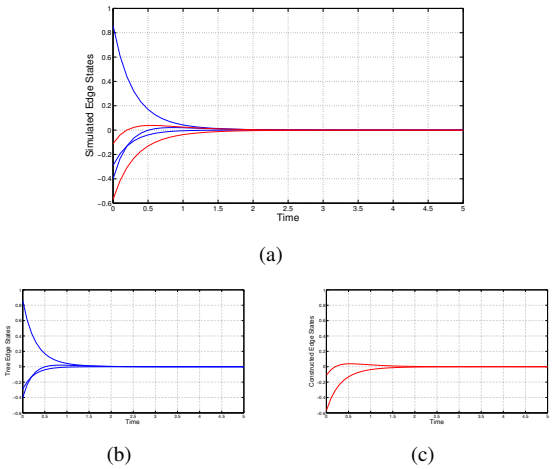

Fig. 5. Edge agreement for the graph of Fig. 2: (a) Simulated edge states for the full order system, (b) Simulated spanning tree edge states, (c) Cycle edge states constructed from the spanning tree edge states

one edge (and consequently a cycle) does not enhance the rate of convergence of the slowest mode.

As discussed in §IV-A, the edge agreement behavior of the entire network can be captured by examining a reduced order system characterized by a spanning tree of the graph. The states of the cycle edges can be reconstructed from the states of the tree edges. To further explore this observation, we first simulated the edge agreement using (20) for the full order graph of Figure 2 as shown in Figure 5(a). We also simulated the reduced order tree edge agreement described in equation (22), and reconstructed the cycle states using the relation presented in (23). Figure 5(b)-(c) depicts the trajectories of the tree edge states and the reconstructed cycle edge states. We note that the combined trajectories in Figure 5(b)-(c) are the same as those shown in Figure 5(a).

VI. Conclusions

In this paper we defined and explored the interpretation of an edge variant of the graph Laplacian in the context of the edge agreement problem. The results presented in this work point to intriguing connections between the well-studied node agreement problem and its edge version. In this direction, we also pointed out strong connections between algebraic and graph-theoretic properties, and their system-theoretic ramifications in the node and edge agreement.

An emerging theme in this paper is the apparent trade-off between the convergence rate of the agreement dynamics, as defined by the smallest non-zero eigenvalue, and the number of edges in the network. We showed how adding cycles to a network can potentially increase the convergence rate to the agreement among the nodes. Furthermore, we showed that one can always represent the edge Laplacian over a graph by a lower dimensional system over its spanning tree subgraph.

References

[1] C. Fall, E. Marland, J. Wagner, and J. Tyson (Editors). *Computational Cell Biology*, Springer, 2005.

[2] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams, *IEEE Transactions on Robotics and Automation*, 14: 926-939, 1998.

[3] B. Bamieh, F. Paganini, and M. Dahleh. Distributed control of spatially-invariant systems, *IEEE Transactions on Automatic Control*, 47 (7): 1091-1107, July 2002.

[4] 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.

[5] G. Walsh, H. Ye, and L. Bushnell. Stability analysis of networked control systems, *American Control Conference*, June 1999.

[6] P. K. C. Wang and F. Y. Hadaegh. Coordination and control of multiple microspacecraft moving in formation, *Journal of the Astronautical Sciences*, 44: 315-355, 1996.

[7] B-D. Chen and S. Lall. Dissipation inequalities for distributed systems on graphs, *IEEE Conference on Decision and Control*, December 2003.

[8] J. Cortes and F. Bullo. Coordination and geometric optimization via distributed dynamical systems, *SIAM Journal on Control and Optimization*, (45) 5: 1543-1574, 2005.

[9] R. D'Andrea and G. E. Dullerud. Distributed control design for spatially interconnected systems, *IEEE Transactions on Automatic Control*, (48) 9: 1478-1495, 2003.

[10] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*, Prentice Hall, 1989.

[11] Y. Hatano and M. Mesbahi. Agreement over random networks, *IEEE Transactions on Automatic Control*, (50) 11: 1867-1872, 2005.

[12] L. Moreau. Stability of multiagent systems with time-dependent communication links, *IEEE Transactions on Automatic Control*, (50) 2: 169 - 182, 2005.

[13] 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.

[14] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer, 2001.

[15] R. A. Horn and C. Johnson. *Matrix Analysis*, Cambridge University Press, 1985.

[16] J. Sandhu, M. Mesbahi, and T. Tsukamaki. Relative sensing networks: observability, estimation, and the control structure, *IEEE Conference on Decision and Control*, December 2005.

[17] M. Fiedler. Algebraic connectivity of graphs, *Czechoslovak Mathematical Journal*, 23 (98): 298-305, 1973.

[18] 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.

[19] 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.