Joint 48th IEEE Conference on Decision and Control and
28th Chinese Control Conference
Shanghai, P.R. China, December 16-18, 2009

$\mathcal{H}_2$ Performance of Agreement Protocol with Noise: An Edge Based Approach

Daniel Zelazo and Mehran Mesbahi†

***Abstract*— This work presents an $\mathcal{H}_2$ performance analysis of the agreement protocol in the presence of noise. The agreement protocol is first transformed into an equivalent system induced by the dynamics of the relative, or edge, system states. The edge based representation is used to perform an $\mathcal{H}_2$ analysis of the system highlighting the roles of cycles in this context. For spanning trees and and certain k-regular graphs, a characterization of the $\mathcal{H}_2$ norm is given in terms of properties of the graph. These results are used to formulate a semi-definite program for sensor selection and placement. Each sensor has an associated cost and fidelity and the developed SDP determines which sensors to use and where to place them such that the $\mathcal{H}_2$ norm of the system, in addition to the sensor costs, are minimized.**

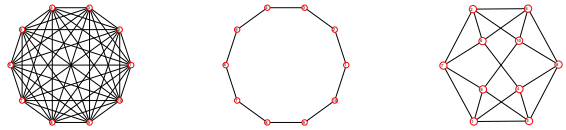
I. Introduction

The linear consensus problem has been extensively studied in the dynamic systems and controls community [6], [9]. Applications of linear consensus include distributed computation algorithms [1], sensor fusion [8], [11], and formation flying [10]. The literature includes many variations of the basic setup including random and switching topologies, stochastic versions, and state-dependent versions [3], [4], [5], [12] . The common analytic theme in all of these approaches relates to the stability and convergent properties of the underlying system.

While these properties describe fundamental aspects for this protocol, they still lack the kind of analysis that is common for more general dynamic systems. For example, the notion of the $\mathcal{H}_2$ performance of the consensus protocol in the context of analysis and synthesis has yet to be examined. This work aims to address this missing systems theoretic component while also providing strong connections between classic systems results and graph theoretic properties.

At the heart of the analysis in this paper lies the edge variant of the consensus problem. Considering the consensus problem from the perspective of states defined over the edges of a graph leads to a transformed system built around an edge variant of the the graph Laplacian, which we term the *Edge Laplacian* [13], [14]. A distinct advantage of studying these problems from an edge perspective is that the dominating dynamics - orthogonal to the agreement space - can be isolated while maintaining strong algebraic properties of the graph in the state matrix.

In this direction, we develop an input-output model based on the edge variant of the consensus protocol. We introduce noise as an exogenous input and consider the $\mathcal{H}_2$ performance of the system as a function of the underlying connection topology. The effects of different topologies are examined, emphasizing analytic results for certain classes of graphs including spanning trees and k-regular graphs. The role of cycles in terms of the $\mathcal{H}_2$ performance is also examined. These results are then applied to develop a synthesis procedure for choosing sensors for the consensus protocol. A semi-definite program is developed that determines where sensors of varying fidelity should be placed in the underlying network to achieve the best $\mathcal{H}_2$ performance.



(a) K_{10} Graph (b) C_{10} Graph (c) 4-regular Graph

Fig. 1. Example of 3 k-regular graphs

II. Preliminaries: Graphs and the Edge Laplacian

An undirected (simple) graph $\mathcal{G}$ is specified by the vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ whose elements characterize the incidence relation between distinct pairs of elements of $\mathcal{V}$. The cardinalities of the vertex and edge sets of $\mathcal{G}$ will be denoted by $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. A *subgraph* of $\mathcal{G}$ 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 $e_k \in \mathcal{E}$ 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. The $|\mathcal{V}| \times |\mathcal{E}|$ 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 $[E(\mathcal{G})]_{ik}$ has the value '+1' if node i is the initial node of edge e_k, '-1' if it is the terminal node, and '0' otherwise.

From the definition of the incidence matrix it follows that the null space of its transpose, $\mathcal{N}(E(\mathcal{G})^T)$, contains the subspace $\mathbf{span}\,\{\mathbf{1}\}$, where $\mathbf{1}$ is the vector with all entries equal to one. The rank of the incidence matrix depends only on $|\mathcal{V}|$ and the number of its connected components, c, with $\mathbf{rank}\, E(\mathcal{G}) = |\mathcal{V}| - c$ [2].

The degree of a vertex $v_i \in \mathcal{V}$, d_i, is the cardinality of the set of vertices adjacent to it. The degree matrix of $\mathcal{G}$, $\Delta(\mathcal{G})$, is a diagonal matrix with the degree of vertex i at its (i, i) position.

The *complete graph* on n nodes, K_n, is the graph where all possible pairs of vertices are adjacent, or equivalently, if the degree of all vertices is $|\,\mathcal{V}\,| - 1$. Figure 1(a) shows the

†The authors are with the Department of Aeronautics and Astronautics, University of Washington, Box 352400, Seattle, WA 98195-2400. Emails: *dlz@aa.washington.edu* and *mesbahi@aa.washington.edu*

978-1-4244-3872-3/09/$25.00 ©2009 IEEE

complete graph on 10 nodes, K_{10}. When every node in a graph has the same degree k, it is called a k-regular graph. The k-regular graph for $k = 2$ is also called the *cycle graph*, C_n. Figure 1(b) and 1(c) shows the cycle graph C_{10} and a 4-regular graph.

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 $|\mathcal{V}|$ vertices with $|\mathcal{V}| - 1$ edges.

Any connected graph $\mathcal{G}$ can be written as the union of two edge-disjoint subgraphs, $\mathcal{G} = \mathcal{G}_\tau \cup \mathcal{G}_c$, where $\mathcal{G}_\tau$ is a spanning tree subgraph and $\mathcal{G}_c$ contains the remaining edges that necessarily complete the cycles in $\mathcal{G}$. Similarly, the columns of the incidence matrix for the graph $\mathcal{G}$ can always be permuted such that $E(\mathcal{G})$ can be written as

$$E(\mathcal{G}) \quad = \quad \left[\begin{array}{cc} E(\mathcal{G}_\tau) & E(\mathcal{G}_c) \end{array} \right]. \tag{1}$$

For notational simplicity, we will use E_τ and E_c to represent the incidence matrix for the tree subgraph and cycle subgraph respectively.

The cycle edges can be constructed from linear combinations of the tree edges, as

$$E_\tau T_\tau^c \quad = \quad E_c, \tag{2}$$

where $T_\tau^c = (E_\tau^T E_\tau)^{-1} E_\tau^T E_c$. Using (2) we obtain the following alternative representation of the incidence matrix.

$$E(\mathcal{G}) \quad = \quad E_\tau \left[\begin{array}{cc} I & T_\tau^c \end{array} \right] \; = \; E_\tau R(\mathcal{G}). \tag{3}$$

In the language of [2], we note that the rows of $R(\mathcal{G})$ forms a basis for the *cut space* of $\mathcal{G}$, and the matrix $\left[\begin{array}{cc} -T_\tau^c & I \end{array} \right]^T$ forms a basis for the *flow space*;

$$\mathbf{IM}\left\{E(\mathcal{G})^T\right\} = \left[\begin{array}{c} I \\ (T_\tau^c)^T \end{array} \right], \qquad \mathbf{Ker}\left\{E(\mathcal{G})\right\} = \left[\begin{array}{c} -T_\tau^c \\ I \end{array} \right].$$

The matrix $R(\mathcal{G})$, which will play an important role in the present work, has a close connection with a number of structural properties of the underlying network. For example, the number of spanning trees in a graph, $\tau(\mathcal{G})$, can be determined from the cut space basis [2], as

$$\tau(\mathcal{G}) \quad = \quad \mathbf{det}\big[R(\mathcal{G})R(\mathcal{G})^T\big]. \tag{4}$$

The graph Laplacian of an oriented graph is defined as

$$L(\mathcal{G}) \quad := \quad E(\mathcal{G})\, E(\mathcal{G})^T \tag{5}$$

which is independent of a particular orientation of the graph. The graph Laplacian of $\mathcal{G}$ is a rank deficient positive semi-definite matrix. The eigenvalues are real and will be ordered and denoted as $0 = \lambda_1(\mathcal{G}) \leq \lambda_2(\mathcal{G}) \leq \ldots \leq \lambda_{|\mathcal{V}|}(\mathcal{G})$.

The *Edge Laplacian* is a variant of the graph Laplacian [13] and is defined as

$$L_e(\mathcal{G}) \quad := \quad E(\mathcal{G})^T\, E(\mathcal{G}). \tag{6}$$

The non-zero eigenvalues of $L_e(\mathcal{G})$ are equivalent to those of $L(\mathcal{G})$, and each cycle in $\mathcal{G}$ corresponds to an eigenvalue at 0 in $L_e(\mathcal{G})$.

Theorem 2.1 ([14]): The graph Laplacian for a connected graph, $L(\mathcal{G})$, is similar to

$$\left[\begin{array}{cc} L_e(\mathcal{G}_\tau)R(\mathcal{G})R(\mathcal{G})^T & 0 \\ 0 & 0 \end{array} \right],$$

where $\mathcal{G}_\tau$ is a spanning tree subgraph of $\mathcal{G}$, and the matrix $R(\mathcal{G})$ is defined in (3).

Proof: We define the transformation matrix

$$S \quad = \quad \left[\begin{array}{cc} E_\tau (E_\tau^T E_\tau)^{-1} & V \end{array} \right], \tag{7}$$

where E_τ is the incidence matrix of $\mathcal{G}_\tau$ and the columns of V are a basis for the null space of $L(\mathcal{G})$, e.g., the vector $\mathbf{1}$. The matrix S is non-singular; in fact its inverse is

$$S^{-1} \quad = \quad \left[\begin{array}{c} E_\tau^T \\ (1/|\mathcal{V}|)\, \mathbf{1}^T \end{array} \right]. \tag{8}$$

Applying the transformation matrix as

$$\begin{aligned} S^{-1} L(\mathcal{G}) S \quad &= \quad \left[\begin{array}{c} E_\tau^T E_\tau \\ 0 \end{array} \right] R(\mathcal{G})R(\mathcal{G})^T \left[\begin{array}{cc} I & 0 \end{array} \right] \\ &= \quad \left[\begin{array}{cc} L_e(\mathcal{G}_\tau)R(\mathcal{G})R(\mathcal{G})^T & 0 \\ 0 & 0 \end{array} \right], \end{aligned} \tag{9}$$

leads to the desired result. ■

Note that when $\mathcal{G} = \mathcal{G}_\tau$ (no cycles), then $R(\mathcal{G}) = I$ and we see a tight connection between the graph and edge Laplacians. Furthermore, for a connected graph $\mathcal{G}_\tau$, the edge Laplacian is guaranteed to be invertible as all its eigenvalues are strictly positive.

III. Graph-theoretic Performance Bounds for Consensus

The noise-free consensus problem is comprised of a collection of n first order dynamic systems of the form

$$\dot{x}_i(t) \quad = \quad u_i(t). \tag{10}$$

The dynamic evolution of each agent is coupled through the control input $u_i(t)$ which is defined to be the sum of the differences between states of an individual unit and its neighbors,

$$u_i(t) \quad = \quad \sum_{j \in \mathcal{N}(i)} (x_j(t) - x_i(t)), \tag{11}$$

where $\mathcal{N}(i)$ denotes the set of agents that are neighbors of agent i, as defined by the connection topology $\mathcal{G}$. Expressing the dynamic evolution of the resulting system in a compact matrix form with $x(t)^T = \left[\begin{array}{ccc} x_1(t), \ldots, x_n(t) \end{array} \right]^T$, we arrive at the following autonomous system,

$$\dot{x}(t) \quad = \quad -L(\mathcal{G})x(t). \tag{12}$$

Agreement of the system (12) is an asymptotic property where each state approaches the same value. For connected graphs, we have that $\lim_{t\to\infty} x(t) = (1/n)Jx(0)$, where J is the matrix of all ones [7].

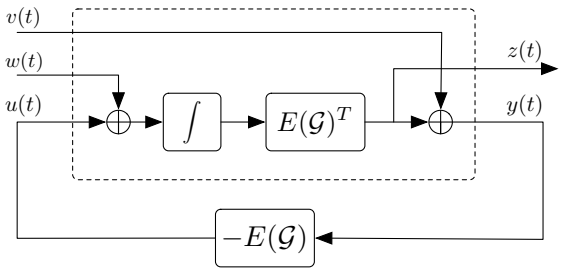


Fig. 2. Open-loop consensus system with output feedback

We now consider a general scenario where noise is introduced at both the process and measurement levels of the consensus protocol. Equation (10) is first modified to include the process noise for each agent, as

$$x_i(t) \quad = \quad u_i(t) + w_i(t). \tag{13}$$

We assume that $w_i(t)$ is a zero-mean white Gaussian noise with covariance $\mathbf{E}[w(t)w(t)^T] = \sigma_w^2 I$.

The measurement is also corrupted by noise, as

$$y(t) \quad = \quad E(\mathcal{G})^T x(t) + v(t). \tag{14}$$

Here, $v(t) \in \mathbb{R}^{|\mathcal{E}|}$ is also a zero-mean white Gaussian noise with covariance $\mathbf{E}[v(t)v(t)^T] = \sigma_v^2 I$.

Equations (13) and (14) can be considered the *open-loop consensus model*. We denote the open-loop system as Σ_{ol},

$$\Sigma_{ol} \quad : \quad \left\{ \begin{array}{rcccc} \dot{x}(t) & = & u(t) & + & w(t) \\ y(t) & = & E(\mathcal{G})^T x(t) & + & v(t) \end{array} \right. . \tag{15}$$

When the output-feedback control $u(t) = -E(\mathcal{G})y(t)$ is applied, the system leads to a generalized consensus protocol with noise. The noisy consensus model will be referred to as the Σ model;

$$\Sigma : \begin{cases} \dot{x}(t) = -L(\mathcal{G})x(t) + \left[\begin{array}{cc} I & -E(\mathcal{G}) \end{array} \right] \left[\begin{array}{c} w(t) \\ v(t) \end{array} \right] \\ z(t) = E(\mathcal{G})^T x(t) \end{cases} . \tag{16}$$

Here, the variable $z(t)$ is introduced as a performance signal to monitor. Note that as $x(t) \to (1/N)Jx(0)$ we have that $z(t) \to \mathbf{0}$. The open-loop system is shown in Figure (2) with the consensus output-feedback law.

The first limiting factor for an $\mathcal{H}_2$ analysis is that for any connected graph, the system Σ has an unbounded $\mathcal{H}_2$ norm due to the presence of the 0 eigenvalue. Therefore, we can consider the performance analysis on the subspace orthogonal to the $\mathbf{1}$ vector, the eigenvector associated with the 0 eigenvalue.

In this direction, we introduce the coordinate transformation $S\hat{x}(t) = x(t)$, where S is defined in (7). Applying this transformation yields

$$\hat{\Sigma} : \begin{cases} \dot{\hat{x}}(t) = \begin{bmatrix} -L_e^\tau R R^T & 0 \\ 0 & 0 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} E_\tau^T & -L_e^\tau R \\ \frac{1}{N}\mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \\ z(t) = \left[\begin{array}{cc} R^T & \mathbf{0} \end{array} \right] \hat{x}(t) \end{cases} . \tag{17}$$

We use the shorthand notation $L_e^\tau = L_e(\mathcal{G}_\tau)$ and $R = R(\mathcal{G})$. The benefit of such a transformation is the algebraic structure of the underlying connection topology is preserved through the Edge Laplacian. Furthermore, we note that the new state $\hat{x}(t)$ can be partitioned as $\hat{x}^T(t) = \left[\begin{array}{cc} x_\tau^T(t) & x_{\mathbf{1}}^T(t) \end{array} \right]^T$ where $x_\tau(t)$ represents the relative state information across the edges of a spanning tree of $\mathcal{G}$, and $x_{\mathbf{1}}(t)$ is the mode in the $\mathbf{1}$ subspace, which corresponds to an unobservable mode of the system.

We can now consider the truncated system containing only the states $x_\tau(t)$ for analysis considerations; in fact, the truncated system represents a minimal realization of (16). We refer to this as the Σ_τ system;

$$\Sigma_\tau : \begin{cases} \dot{x}(t) = -L_e^\tau R R^T x_\tau(t) + \sigma_w E_\tau^T \hat{w}(t) - \sigma_v L_e^\tau R \hat{v}(t) \\ z(t) = R^T x_\tau(t) \end{cases} . \tag{18}$$

The signals $\hat{w}(t)$ and $\hat{v}(t)$ are the normalized process and measurement noise signals. The performance variable, $z(t)$, contains information on the tree states in addition to the cycle states. Here we recall that the cycle states are linear combination of the tree states and we note that $z(t)$ actually contains redundant information. This is highlighted by recognizing that the tree states converging to the origin forces the cycle states to do the same. Consequently, we will consider the system with cycles as well as a system containing only the tree states at the output, which we denote as $\hat{\Sigma}_\tau$. This also allows for a means to quantify the affect of cycles on the performance.

$$\hat{\Sigma}_\tau : \begin{cases} \dot{x}(t) = -L_e^\tau R R^T x_\tau(t) + \sigma_w E_\tau^T \hat{w}(t) - \sigma_v L_e^\tau R \hat{v}(t) \\ z(t) = x_\tau(t) \end{cases} . \tag{19}$$

The $\mathcal{H}_2$ norm of Σ_τ and $\hat{\Sigma}_\tau$ can be calculated as

$$\|\Sigma_\tau\|_2^2 = \mathbf{Tr}[R^T X^* R], \qquad \|\hat{\Sigma}_\tau\|_2^2 = \mathbf{Tr}[X^*], \tag{20}$$

where X^* is the positive-definite solution to the Lyapunov equation

$$-L_e^\tau R R^T X^* - X^* R R^T L_e^\tau + \sigma_w^2 L_e^\tau + \sigma_v^2 L_e^\tau R R^T L_e^\tau = 0. \tag{21}$$

The structure of (21) suggests that any solution be dependent on certain properties of the graph. In fact, the solution can be written by inspection by noting that

$$\sigma_w^2 L_e^\tau + \sigma_v^2 L_e^\tau R R^T L_e^\tau = L e^\tau \left(\sigma_w^2 (L_e^\tau)^{-1} + \sigma_v^2 R R^T \right) L_e^\tau.$$

The solution to (21) is

$$X^* \quad = \quad \frac{1}{2} \left(\sigma_w^2 (R R^T)^{-1} + \sigma_v^2 L_e^\tau \right), \tag{22}$$

and we arrive at the following result.

Theorem 3.1: The $\mathcal{H}_2$ norm of the Σ_τ system is

$$\|\Sigma_\tau\|_2^2 \quad = \quad \frac{\sigma_w^2}{2}(n-1) + \sigma_v^2 |\mathcal{E}|. \tag{23}$$

The $\mathcal{H}_2$ norm of the $\hat{\Sigma}_\tau$ system is

$$\|\hat{\Sigma}_\tau\|_2^2 \quad = \quad \frac{\sigma_w^2}{2} \mathbf{Tr}[(R R^T)^{-1}] + \sigma_v^2 (n-1). \tag{24}$$

Proof: The proof follows from (22) and noting that $\mathbf{Tr}[L_e^\tau] = 2(n-1)$, or twice the number of edges in a spanning tree. ∎

We note that $\|\Sigma_\tau\|_2^2$ is a linear function of the number of edges in the graph. This has a clear physical interpretation, as the addition of each edge corresponds to an amplification of the noises.

While a general graph theoretic characterization of (24) may be hard to derive, certain graph structures allow for complete characterizations of the solution, which we present below.

A. Spanning Trees

The first case resulting in a simplification of (23) arises when $\mathcal{G}$ is a spanning tree. In this case $R = I$ and (23) simplifies to

$$\|\hat{\Sigma}_\tau\|_2^2 \;=\; (n-1)\left(\frac{\sigma_w^2}{2}+\sigma_v^2\right) \tag{25}$$

An interesting consequence of this result is that *all* spanning trees result in the same system performance. That is, the choice of spanning tree (e.g. a path or a star) does not affect the performance. As expected, in this scenario $\|\Sigma_\tau\|_2^2 = \|\hat{\Sigma}_\tau\|_2^2$. We also note that when the noises have different covariance values, this analysis becomes significantly more complicated, and we consider this in §IV.

B. k-Regular Graphs

Regular graphs also lead to a simplification of (24). Any connected k-regular graph will contain cycles resulting in a non-trivial expression for RR^T. The $\mathcal{H}_2$ norm is therefore intimately related to the cut space of the graph. A direct characterization of RR^T in terms of basic properties of the graph is challenging for arbitrary k-regular graphs. However, certain k-regular graphs leads to further simplifications, as presented below.

Denote the eigenvalues of RR^T by μ_i and note that

$$\mathbf{Tr}\big[(RR^T)^{-1}\big] \;=\; \sum_{i=1}^{n-1}\frac{1}{\mu_i} \;=\; \frac{1}{\tau(\mathcal{G})}\sum_{i=1}^{n-1}\prod_{j\neq i}^{n-1}\mu_j \tag{26}$$

The quantity $\prod_{j\neq i}^{n-1}\mu_j$ is recognized as a first minor of the matrix RR^T.

Lemma 3.2: The k-regular graph with degree 2 (cycle graph) has n spanning trees and

$$\mathbf{Tr}\big[(RR^T)^{-1}\big] \;=\; \frac{(n-1)^2}{n}. \tag{27}$$

The $\mathcal{H}_2$ norm of the $\hat{\Sigma}_\tau$ system when the underlying graph is the cycle graph C_n is given as

$$\|\hat{\Sigma}_2\|_2^2 \;=\; (n-1)\left(\frac{\sigma_w^2(n-1)}{n}+\sigma_v^2\right). \tag{28}$$

Proof: Without loss of generality, we consider a directed path graph on n nodes, with initial node v_1 and terminal node v_n as the spanning tree subgraph G_τ. Index the edges as $e_i = (v_i, v_{i+1})$. The cycle graph is formed by adding the edge $e_n = (v_n, v_1)$. For this graph, we have $T_\tau^c = \mathbf{1}_{n-1}$ and $RR^T = I + J_{n-1}$. From this it follows that $\mathbf{det}\big[RR^T\big] = N$ and all its first minors have value $N-1$. Combined with (23) yields the desired result. ∎

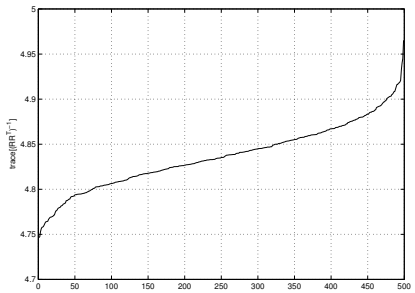


Fig. 3. $\mathbf{Tr}[(RR^T)^{-1}]$ for random 5-regular graphs

Lemma 3.3: The k-regular graph with degree $n-1$ (complete graph) has n^{n-2} spanning trees and

$$\mathbf{Tr}\big[(RR^T)^{-1}\big] \;=\; \frac{2(n-1)n^{n-3}}{n^{n-2}} = \frac{2(n-1)}{n}. \tag{29}$$

The $\mathcal{H}_2$ norm of the $\hat{\Sigma}_\tau$ system when the underlying graph is the complete graph K_n is given as

$$\|\hat{\Sigma}_\tau\|_2^2 \;=\; (n-1)\left(\frac{\sigma_w^2}{n}+\sigma_v^2\right). \tag{30}$$

Proof: Without loss of generality, we consider a star graph with center at node v_1 and all edges are of the form $e_k = (v_1, v_{k+1})$. Then the cycles in the graph are created by adding the edges $e = (v_i, v_j),\ i,j \neq 1$ and $RR^T = nI - J_{n-1}$. It then follows that $\mathbf{det}\big[RR^T\big] = n^{n-2}$ and all the first minors have value $2n^{n-3}$. Combined with (23) yields the desired result. ∎

In general, one expects the system norm to decrease as the regularity increases. However, it is not clear how the cycle structure directly affects the norm. To illustrate this, 500 random regular graphs of degree 5 were generated in MATLAB. For each instance, the value $\mathbf{Tr}[(RR^T)^{-1}]$ was calculated, sorted, and plotted in Figure 3. In this example, although the degree of each node remains constant, the actual cycle structure, meaning both the number of independent cycles and the length of those cycles, varies greatly.

The above analysis suggests that other classes of graphs may exist that lead to a simplification of the expression in (23). Ultimately, any such simplification will relate to the matrix $(RR^T)^{-1}$, which is intimately related to the cycles of the graph.

C. Cycle Contributions

Using the above analysis we can begin to quantitatively understand how cycles affect the $\mathcal{H}_2$ performance. For example, examining the ratio

$$\frac{\|\Sigma_\tau(\mathcal{G})\|_2^2}{\|\Sigma_\tau(\mathcal{G}_\tau)\|_2^2}$$

can give a good indication of how the cycles increase the $\mathcal{H}_2$ norm. Recall that $\mathcal{G}$ is in general a graph containing cycles and $\mathcal{G}_\tau \subseteq \mathcal{G}$ is the spanning tree subgraph.

For example, consider the cycle graph C_n and assume unit covariance for both the process and measurement noises. The ratio becomes

$$\frac{\|\Sigma_\tau(C_n)\|_2^2}{\|\Sigma_\tau(\mathcal{G}_\tau)\|_2^2} \quad = \quad \frac{3n-1}{3(n-1)}. \tag{31}$$

In this case, we note that as the number of nodes increases the effect of the cycle (there is only 1) decreases and becomes negligible in the limit.

Similarly, for the complete graph K_n we have

$$\frac{\|\Sigma_\tau(K_n)\|_2^2}{\|\Sigma_\tau(\mathcal{G}_\tau)\|_2^2} \quad = \quad \frac{n+1}{3}. \tag{32}$$

Here we see that the norm is amplified linearly as a function of the number of nodes in the graph. It is worth mentioning here that typical performance measures for consensus problems, such as $\lambda_2(\mathcal{G})$, would favor the complete graph over the cycle graph. However, in terms of the $\mathcal{H}_2$ performance we see that there is a penalty to be paid for faster convergence.

Alternatively, insight is also gained by considering the ratio

$$\frac{\|\Sigma_\tau(\mathcal{G})\|_2^2}{\|\hat{\Sigma}_\tau(\mathcal{G})\|_2^2},$$

which highlights the effects of including cycles in the performance variable $z(t)$.

For the cycle graph we have

$$\frac{\|\Sigma_\tau(C_n)\|_2^2}{\|\hat{\Sigma}_\tau(C_n)\|_2^2} \quad = \quad \frac{n(3n-1)}{2(n-1)(2n-1)}. \tag{33}$$

Here we note that as $n \to \infty$ the ratio approaches 0.75, which suggests that the effect of including the cycle for performance evaluation does not vary significantly with the size of the graph.

For the complete graph we have

$$\frac{\|\Sigma_\tau(K_n)\|_2^2}{\|\hat{\Sigma}_\tau(K_n)\|_2^2} \quad = \quad \frac{n}{2}. \tag{34}$$

As with (32) we see that the inclusion of cycles results in a linear function of the number of nodes in the graph.

IV. Sensor Placement with $\mathcal{H}_2$ Performance

In this section we consider the problem of sensor selection and placement for consensus in the context of its $\mathcal{H}_2$ performance. Consider, for example, a scenario where there are two types of sensors available for the relative measurements in the open-loop consensus problem. One sensor is a high-fidelity and high cost sensor, with associated noise covariance $\overline{\sigma}_v^2$. The other sensor is cheaper with covariance $\underline{\sigma}_v^2 > \overline{\sigma}_v^2$. When synthesizing the topology for the consensus problem, the designer must consider the tradeoff between the sensor costs and the system performance.

In this direction, we consider a modification of the system in (18),

$$\Sigma_\tau : \begin{cases} \dot{x}(t) \;=\; -L_e^\tau R R^T x_\tau(t) + \sigma_w E_\tau^T \hat{w}(t) - \\ \qquad\qquad\qquad\qquad L_e^T R \Gamma \hat{v}(t) \\ z(t) \;=\; R^T x_\tau(t) \end{cases}, \tag{35}$$

where $\hat{w}(t)$ and $\hat{v}(t)$ are the normalized noise signals. The matrix Γ is a diagonal matrix with elements σ_i corresponding to the variance of the sensor on edge i.

The most general version of this problem considers a finite set of p sensors each with an associated variance,

$$P \quad = \quad \{\sigma_i^2, i = 1, 2, \ldots, p\}. \tag{36}$$

For each element $\sigma_i^2 \in P$ there is an associated cost $c(\sigma_i^2)$. The cost function has the property that $c(\sigma_i^2) > c(\sigma_j^2)$ if $\sigma_i^2 < \sigma_j^2$. The mixed-integer program can then be written as

$$\begin{aligned} &\mathcal{P}_1 \\ &\min_{X,W} \quad \lambda \mathbf{Tr}[R^T X R] + \sum_{i=1}^{|\mathcal{E}|} c(w_i) \\ &\text{s.t.} \quad W = \mathbf{diag}\{w_1, \ldots, w_{|\mathcal{E}|}\}, \\ &\qquad w_i \in P, \\ &\qquad \sum_i w_i \leq \overline{\sigma}, \\ &-L_e^T R R^T X - X R R^T L_e^T + \sigma_w^2 L_e^\tau + L_e^T R W R^T L_e^\tau \;=\; 0, \end{aligned} \tag{37}$$

where λ represents a weighting on the $\mathcal{H}_2$ performance of the solution, and $\overline{\sigma}$ represents the maximum allowable aggregate noise. Note that in general $|\mathcal{E}| \min_i \sigma_i^2 \leq \overline{\sigma} \leq |\mathcal{E}| \max_i \sigma_i^2$.

The problem $\mathcal{P}_1$ is combinatorial in nature, as a discrete decision must be made as to which sensor to use and where to place it in the network. While a tractable solution algorithm is still under investigation, certain relaxations can be made that lead to a more approachable problem. The first relaxation removes the discrete nature of the set P into a simple box-type constraint, as

$$\hat{P} \quad = \quad \left[\underline{\sigma}^2,\, \overline{\sigma}^2\right]. \tag{38}$$

The cost function now can be written as a continuous map $c : \hat{P} \mapsto \mathbb{R}$ that is a covex and strictly decreasing function. The simplest version of such a function would be the linear map $c(\sigma_i^2) = -\beta\sigma_i^2$ for some $\beta > 0$. These relaxations lead to the following modified program, $\mathcal{P}_2$;

$$\begin{aligned} &\mathcal{P}_2 \\ &\min_{X,W} \quad \lambda \mathbf{Tr}[R^T X R] - \beta \mathbf{Tr}[W] \\ &\text{s.t.} \quad W = \mathbf{diag}\{w_1, \ldots, w_{|\mathcal{E}|}\}, \\ &\qquad \underline{\sigma}^2 \leq w_i \leq \overline{\sigma}^2, \\ &\qquad \sum_i w_i \leq \overline{\sigma}, \\ &-L_e^T R R^T X - X R R^T L_e^T + \sigma_w^2 L_e^\tau + L_e^T R W R^T L_e^\tau \;=\; 0. \end{aligned} \tag{39}$$

As in the analysis of §III, we would expect that certain topologies lead to a simplification in the above programs. As

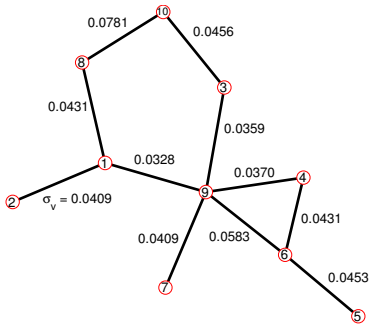


Fig. 4. A graph on 10 nodes with optimal sensor selection

an example of $\mathcal{P}_2$, we consider the sensor selection for the graph in Figure 4. A random graph on 10 nodes with an edge probability of 0.15 was generated. The graph is connected and contains two independent cycles, resulting in the most general problem instance. The sensor constraints used are $\hat{P} = [0.001,\ 0.1]$ and $\overline{\sigma}^2 = 0.501$. Finally, the cost function tuning values were chosen as $\beta = 5$ and $\lambda = 1$. It is worth mentioning that the selection of these values is currently a trial and error process. Ideally, they should be chosen in such a way so that one term of the objective function does not overly dominate the other.

Solving $\mathcal{P}_2$ resulted in a non-trivial selection of sensors for each edge. The sensor covariance for each edge is labeled in Figure 4. It is interesting to note that the highest fidelity sensors tend to be concentrated around the node of highest degree. Also, the edge with the lowest fidelity sensor is furthest away from the node of highest degree. It seems rather intuitive to place the lower fidelity sensors in "low traffic" areas. That is, edges that are adjacent to a low number of other edges.

V. Conclusions

This work presented an $\mathcal{H}_2$ performance analysis of the consensus protocol with both process and measurement noises present. In order to perform such an analysis, the consensus model was transformed into an edge representation leading to a minimal realization of the system. While the traditional consensus problem is centered around analysis of the graph Laplacian, the edge variant relies on the Edge Laplacian. An advantage of this transformation is the ability to do analysis of the system orthogonal to the agreement subspace while preserving the strong algebraic properties of the graph via the Edge Laplacian.

The $\mathcal{H}_2$ performance of the consensus protocol was shown to be dependent on the number of edges in the graph. Consequently, this suggests that cycles play an important role in terms of the system performance by way of noise propagation. This is in stark contrast to traditional analysis of consensus problems which focuses on the convergence rate of the system. It is well understood that an increase in the number of edges in the graph corresponds to an increase in the second smallest eigenvalue of the graph Laplacian, resulting in faster convergence. However, in this context, the addition of edges produce an adverse affect in performance.

The analysis results of §III showed that certain classes of graphs - spanning trees and k-regular graphs - result in a graph theoretic description of the system norm. Certainly, it seems that there should be other classes of graphs that allow for similar simplifications, and this is currently under investigation. The observations on the performance ratio relating to the cycle graph suggest that some graphs scale better than others in terms of performance and the number of nodes.

These results were then applied to an optimization problem aimed at selecting sensors for a fixed topology consensus problem. Although the complete statement of the problem is numerically challenging, the relaxation provides a reasonable approach to this problem. This formulation can naturally be pushed further to include, for example, topology design with both favorable convergence rates, good $\mathcal{H}_2$ performance, and low sensor costs. More subtly, it highlights a connection between combinatorial optimization problems and synthesis methods for networked dynamic systems. Consequently, the development of numerically tractable algorithms for the design of these systems is an essential component for the maturation of this field.

Acknowledgements

The research of the authors was sponsored by NSF grant ECCS-0501606.

References

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*, Prentice Hall, 1989.
[2] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer, 2001.
[3] Y. Hatano, A. Das, and M. Mesbahi. Agreement in Presence of Noise: Pseudogradients on Random Geometric Networks, *44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.
[4] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Transactions on Automatic Control*, (50) 11: 1867-1872, 2005.
[5] M. Mesbahi. "State-dependent graphs," *IEEE Conference on Decision and Control*, December 2003.
[6] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and Cooperation in Networked Multi-Agent Systems, *Proceedings of the IEEE*, (95) 1: 1-17, 2007.
[7] 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.
[8] R. Olfati-Saber and J. S. Shamma. Consensus Filters for Sensor Networks and Distributed Sensor Fusion, *IEEE Conference on Decision and Control*, Seville, Spain, December 2005.
[9] W. Ren, R. Beard, and E. Atkins. Information Consensus in Multivehicle Cooperative Control, *IEEE Control Systems Magazine*, April 2007.
[10] W. Ren. Consensus Strategies for Cooperative Control of Vehicle Formations, *IET Control Theory & Applications* (1) 2: 505-512, 2007.
[11] D. Spanos and R. Murray. Distributed Sensor Fusion Using Dynamic Consensus *2005 IFAC World Congress*, Prague, Czech Repulic, July 2005.
[12] L. Xiao, S. Boyd, and S. Kim. Distributed Average Consensus with Least-Mean-Square Deviation, *Journal of Parallel and Distributed Computing*, (67) 1: 33-46, 2007.
[13] D. Zelazo, A. Rahmani, and M. Mesbahi, Agreement on Edges via the Edge Laplacian, *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007.
[14] D. Zelazo and M. Mesbahi. Edge agreement: Graph-theoretic performance bounds and passivity analysis, *IEEE Transactions on Automatic Control* 2009 (submitted).