2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Graph Weight Design for Laplacian Eigenvalue Constraints with Multi-Agent Systems Applications

S. Yusef Shafi, Murat Arcak, Laurent El Ghaoui

Abstract— **We adjust the node and edge weightings of graphs using convex optimization to impose bounds on their Laplacian spectra. We first derive necessary and sufficient conditions that characterize the feasibility of spectral bounds given positive node and edge weightings. Next, we propose algorithms that exploit convexity to achieve these bounds. The design and analysis tools are useful for a variety of stability and control problems in multi-agent systems.**

I. Introduction

A well-studied tool for characterizing the interconnection topology of a networked multi-agent system is the graph Laplacian matrix [1]. In particular, the spectrum of the Laplacian contains useful information about the dynamics of the network. For example, the smallest positive eigenvalue of a Laplacian, known as the algebraic connectivity, or Fiedler eigenvalue [2], is a common measure of how well connected a network is [3][4][5][6]. On the other hand, the largest eigenvalue must be sufficiently small for stability of discrete-time consensus algorithms [4][7], and for continuous-time formation control algorithms when agent dynamics can be destabilized by high gain feedback [8].

We present a scheme to enforce constraints on the Laplacian spectrum by treating both node and edge weights as decision variables. Let λ_i be the ith-smallest eigenvalue of the Laplacian, whose eigenvalues are ordered from least to greatest. Given $m \in \{2, \ldots, n\}$ and $\underline{\lambda}_m > 0$, the lower eigenvalue bound assignment problem is to guarantee $\lambda_m \geq \underline{\lambda}_m$. Likewise, given $p \in \{2, \ldots, n\}$ and $\overline{\lambda}_p > 0$, the upper eigenvalue bound assignment problem is to guarantee $\lambda_p \leq \overline{\lambda}_p$. Our goal is to achieve individual upper and lower bounds for several Laplacian eigenvalues simultaneously. We show how these bounds can be recast as linear matrix inequality constraints [9] that can be applied using semidefinite programming.

Convex optimization solutions to several graph problems are well-documented in the literature, including fastest distributed linear averaging (FDLA) [10], minimization of total effective resistance on a graph [11], fastest mixing Markov chains [12] and processes [13], and Fiedler eigenvalue maximization through vertex positioning [14]. In FDLA [10], a particular interconnection structure for a discrete system with symmetric interconnections is specified. The number of iterations required for linear averaging is minimized by finding a particular weight distribution that assigns iterative update laws for each node's state. The goal in many resistor network problems [11] is to minimize the total effective resistance on a graph by assigning different weights representing resistances to the links connecting the nodes of an electrical network. The aim for fastest mixing Markov chains [12] and processes [13] is to find the optimal transition probabilities between states to reach a stationary distribution as quickly as possible. Finally, vertex positioning [14] aims to find the optimal locations of vertices, corresponding to edge weights, in order to maximize the Fiedler eigenvalue.

Our approach is unique when compared to previous literature on optimization of the Laplacian spectrum because it is applicable to any selection of Laplacian eigenvalues and assigns weights independently to both nodes and edges. In [15], we showed how to adjust node and edge weights to reduce the gap between the largest and smallest positive eigenvalues. In the present paper, we develop a general framework for adjusting several Laplacian eigenvalues and reduce [15] to a special case. We also give a detailed account of applications to formation control problems for multi-agent systems.

The remainder of the paper is organized as follows. Section II introduces mathematical preliminaries that are necessary for our analysis. Section III outlines a general optimization framework that enables upper and lower bounds on several Laplacian eigenvalues simultaneously using node and edge weighting. Section IV presents sample problems that can be formulated and solved using the methods of Section III. Section V explores applications to multi-agent systems.

II. Preliminaries

We review the following results from linear algebra, which we will use in Section III. The first result concerns the eigenvalues of a product of two matrices ([16], Theorem 1.3.20):

Lemma 2.1: Let $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times n}$, and $n \geq m$. Then AB and BA have m identical eigenvalues with AB having $n - m$ additional eigenvalues at zero.

The next lemma follows from the Courant-Fischer theorem, which characterizes the eigenvalues of a symmetric matrix ([16], Corollary 4.3.23):

Lemma 2.2: If $A \in \mathbb{R}^{n\times n}$ is symmetric and if $x^T Ax \geq 0$

The authors are with the department of Electrical Engineering and Computer Sciences, University of California, Berkeley, USA. Email: {yusef,arcak,elghaoui@eecs.berkeley.edu}. Research supported in part by NSF grants ECCS-0852750 and ECCS-1101876 and AFOSR grant FA9550-11-1-0244.

for all vectors $x \in \mathbb{R}^n$ in a k-dimensional subspace, then A has at least k nonnegative eigenvalues.

Definition 2.3: The square matrices A and B are **congruent** if $B = SAS^T$ for some square, nonsingular S.

The following lemma is known as Sylvester's Law of Inertia ([16], Theorem 4.5.8):

Lemma 2.4: Let $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times n}$ be symmetric matrices. A and B are congruent if and only if A and B have the same *inertia*, i.e., the same number of positive, negative, and zero eigenvalues.

An inequality due to Sylvester characterizes the relationship between the eigenvalues of two matrices and their products ([17], Section 3.5):

Lemma 2.5: Given two matrices $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{n\times q}$, the following inequality holds:

$$\begin{aligned}\text{rank}(A) + \text{rank}(B) - n \leq \text{rank}(AB) \\ \leq \min\{\text{rank}(A), \text{rank}(B)\}.\end{aligned} \tag{1}$$

We next review notions from spectral graph theory that are essential to this paper.

A *graph* $G = G(V, E)$ is a collection of nodes V and a corresponding set of edges E. In this paper, we consider *undirected* graphs, where two nodes are *connected* when there exists an edge incident to both. A graph itself is connected if there exists a sequence of edges connecting any pair of nodes in the graph.

Given an undirected graph $G(V, E)$ with n nodes and h edges, an *incidence matrix* $A \in \mathbb{R}^{n\times h}$ is an $n \times h$ matrix, each of whose columns, indexed by $k = 1, \ldots, h$, represents an edge in E linking nodes v_i and v_j in V, with $[A]_{ik} = 1$, $[A]_{jk} = -1$, and $[A]_{lk} = 0$ for all $l \neq i, j$. We note that the incidence matrix is not unique for an undirected graph, and choice of orientation does not change our results. We denote by:

$$L = AA^T \tag{2}$$

the $n \times n$ *nominal Laplacian*, and by:

$$L_e = AKA^T \tag{3}$$

the *edge-weighted Laplacian*, where $K \succeq 0$ is the diagonal *edge weighting matrix*. We denote by:

$$L_g = M^{-1}AKA^T \tag{4}$$

the *node- and edge- weighted graph Laplacian* (henceforth *weighted Laplacian*), where $M \succ 0$ is the diagonal *node weighting matrix*.

We first recall key facts about Laplacian matrices [1]. L and L_e are symmetric positive semidefinite, with at least one eigenvalue at zero corresponding to an eigenvector $\mathbf{1}_n = \frac{1}{\sqrt{n}}[1 \ldots 1]^T$. If the graph represented by L or L_e is connected, then L or L_e, respectively, has exactly one eigenvalue at zero. Although L_g is not symmetric in general, its eigenvalues possess properties similar to those of L and L_e, as shown in the following lemma proven in [15]:

Lemma 2.6: Every eigenvalue of L_g is real and nonnegative. If L_g represents a connected graph, then all eigenvalues of L_g, excepting one at zero, are positive.

III. Convex Characterizations of Upper and Lower Eigenvalue Constraints

Our goal is to find node and edge weighting matrices M and K to assign lower and upper bounds on the spectrum of L_g. We wish to achieve individual bounds for several Laplacian eigenvalues simultaneously. We now formulate this problem as a convex optimization task.

A. Bounding Eigenvalues from Below

Given $m < n$ and $\underline{\lambda}_m > 0$, we wish to design node and edge weights M and K, respectively, such that $\lambda_m(L_g) \geq \underline{\lambda}_m$, where $\lambda_m(L_g)$ denotes the m-th smallest eigenvalue of L_g. To construct a linear matrix inequality enforcing the eigenvalue constraint, we make use of the following lemma:

Lemma 3.1: Suppose that $m < n$, $Q_m \in \mathbb{R}^{n\times(n-m+1)}$ is a full column rank matrix whose columns are orthogonal, and S is a symmetric matrix. If $Q_m^T S Q_m \succeq 0$, then $\lambda_m(S) \geq 0$.

Proof: The result follows immediately from Lemma 2.2: the subspace spanned by the columns of Q_m is $n-m+1$ dimensional, so $\lambda_m(S) \geq 0$. ∎

The next theorem provides a sufficient condition in the form of a linear matrix inequality constraint to enforce lower eigenvalue bounds:

Theorem 3.2: Let Q_m be as in Lemma 3.1. The constraint

$$Q_m^T(L_e - \underline{\lambda}_m M)Q_m \succeq 0 \tag{5}$$

implies that $\lambda_m(L_g) \geq \underline{\lambda}_m$.

Proof: First, we note by Lemma 3.1 that if (5) holds, then the matrix $L_e - \underline{\lambda}_m M$ has at most $m-1$ negative eigenvalues. By congruence, $M^{-1/2}L_eM^{-1/2} - \underline{\lambda}_m I$ has at most $m-1$ negative eigenvalues, which means that the symmetric positive semidefinite matrix

$$L_s \triangleq M^{-1/2}L_eM^{-1/2} \tag{6}$$

has at most $m-1$ eigenvalues less than $\underline{\lambda}_m$. Similarity of L_g to L_s implies that L_g has at most $m-1$ eigenvalues less than $\underline{\lambda}_m$, implying that $\lambda_m(L_g) \geq \underline{\lambda}_m$. ∎

We now present a convex feasibility program that enforces the lower eigenvalue bound sufficient linear matrix inequality condition of Theorem 3.2:

$$\begin{aligned} \text{Find} \quad & M,\ K \\ \text{subject to} \quad & Q_m^T(AKA^T - \underline{\lambda}_m M)Q_m \succeq 0 \\ & M \succ 0 \\ & K \succeq 0 \\ & M,\ K \text{ diagonal.} \end{aligned} \tag{7}$$

We note that the lower eigenvalue bound can also be enforced by scaling M by $\frac{\lambda_m(L)}{\underline{\lambda}_m}$ or K by $\frac{\underline{\lambda}_m}{\lambda_m(L)}$. However,

when the graph optimization problem imposes upper eigenvalue constraints or objective functions, this approach would be highly conservative. In contrast, (7) can be combined with other constraints and objectives without this conservatism.

Theorem 3.2 provides only a sufficient condition to imply $\lambda_m(L_g) \geq \underline{\lambda}_m$, because the choice of Q_m is arbitrary. We now present a necessary and sufficient condition enabled by a specific choice of Q_m:

Theorem 3.3: $\lambda_m(L_g) \geq \underline{\lambda}_m$ if and only if $\mathcal{Q}_m^T(L_e - \underline{\lambda}_m M)\mathcal{Q}_m \succeq 0$, where $\mathcal{Q}_m \in \mathbb{R}^{n\times(n-m+1)}$ is the matrix whose columns are the eigenvectors corresponding to the $n-m+1$ largest eigenvalues of $L_e - \underline{\lambda}_m M$.

Proof: Necessity follows from Theorem 3.2. To prove sufficiency, suppose that $\lambda_m(L_g) \geq \underline{\lambda}_m$. By similarity, $L_s = M^{-1/2}L_eM^{-1/2}$ has the same spectrum as L_g. Then $L_s - \underline{\lambda}_m I$ has at most $m-1$ negative eigenvalues. By congruence, so does $L_e - \underline{\lambda}_m M$. Considering the projection matrix $\mathcal{Q}_m\mathcal{Q}_m^T$, it follows that $(L_e - \underline{\lambda}_m M)\mathcal{Q}_m\mathcal{Q}_m^T$ must have exclusively nonnegative eigenvalues. By Lemma 2.1,

$$\mathcal{Q}_m^T(L_e - \underline{\lambda}_m M)\mathcal{Q}_m \succeq 0. \tag{8}$$

■

Theorem 3.3 is the basis for an iterative procedure presented in Section IV that allows for improved performance when the constraints of (7) are paired with an objective.

B. Bounding Eigenvalues from Above

Given $p \leq n$ and $\overline{\lambda}_p \geq 0$, we wish to design node and edge weights M and K, respectively, such that $\lambda_p(L_g) \leq \overline{\lambda}_p$. We construct a linear matrix inequality enforcing this eigenvalue constraint. The analysis is similar to that of the previous section, and so the proofs are omitted.

Theorem 3.4: Let $U_p \in \mathbb{R}^{n\times p}$ be a full column rank matrix whose columns are orthogonal. The constraint

$$U_p^T(\overline{\lambda}_p M - L_e)U_p \succeq 0 \tag{9}$$

implies that $\lambda_p(L_G) \leq \overline{\lambda}_p$.

We now present a convex feasibility program that enforces the upper eigenvalue bound sufficient linear matrix inequality condition of Theorem 3.4:

$$\begin{array}{rl} \text{Find} & M,\, K \\ \text{subject to} & U_p^T(\overline{\lambda}_p M - AKA^T)U_p \succeq 0 \\ & M \succ 0 \\ & K \succeq 0 \\ & M,\, K \text{ diagonal} \end{array} \tag{10}$$

As in the case of bounding eigenvalues from below, Theorem 3.4 provides only a sufficient condition to imply $\lambda_p(L_g) \leq \overline{\lambda}_p$. The following theorem gives a necessary and sufficient condition enabled by a specific choice of U_p:

Theorem 3.5: $\lambda_p(L_g) \leq \overline{\lambda}_p$ if and only if $\mathcal{U}_p^T(\overline{\lambda}_p M - L_e)\mathcal{U}_p \succeq 0$, where $\mathcal{U}_p \in \mathbb{R}^{n\times p}$ is the matrix whose columns are the eigenvectors corresponding to the p smallest eigenvalues of $\overline{\lambda}_p M - L_e$.

An iterative procedure presented in Section IV employs Theorem 3.5 and allows for improved performance when the constraints of (7) and (10) are paired with an objective.

A special case of Theorem 3.5 is when $p = n$. Since, in this case, the $\mathcal{U}_p$ that satisfies Theorem 3.5 is a square, orthogonal matrix, Theorem 3.5 simplifies to the following corollary:

Corollary 3.6: $\lambda_n(L_g) \leq \overline{\lambda}_n$ if and only if $\overline{\lambda}_n M - L_e \succeq 0$.

IV. Examples of Graph Design Problems

We provide two sample problems that can be addressed by combining (7) and (10). In our numerical examples, we require that all node and edge weights be contained in $[\epsilon, \epsilon^{-1}]$, where $\epsilon < 1$ is a small positive parameter that guarantees that the largest and smallest weights do not have too great a relative difference. We perform our numerical examples using `CVX`, a package for *disciplined convex programming* [18] [19], and the SDPT3 interior point solver [20].

A. Minimizing the Largest Eigenvalue Given a Minimum Connectivity Constraint

In formation control problems (see, e.g., Section V), it is desirable to have a lower bound on λ_2 to ensure adequate convergence time while at the same time imposing an upper bound on λ_n for stability. We present the problem of minimizing the largest eigenvalue $\lambda_n(L_g)$ of a graph given the requirement $\lambda_2(L_g) \geq \underline{\lambda}_2$, making use of (7) and (10) as well as including upper and lower bounds on the entries of M and K:

$$\begin{array}{ll} \underset{\kappa,M,K}{\text{minimize}} & \kappa \\ \text{subject to} & \kappa\underline{\lambda}_2 M - AKA^T \succeq 0 \\ & Q_2^T(AKA^T - \underline{\lambda}_2 M)Q_2 \succeq 0 \\ & \epsilon^{-1}I \succeq M \succeq \epsilon I, \epsilon^{-1}I \succeq K \succeq \epsilon I \\ & M,\, K \text{ diagonal}. \end{array} \tag{11}$$

The problem is quasiconvex for any $Q_2 \in \mathbb{R}^{n\times(n-1)}$. To find the optimal κ for the problem, we perform a bisection on the interval $[\underline{\lambda}_2, \lambda_n(L)]$, where in each iteration, a convex feasibility problem is solved for the value of κ given by the bisection. As discussed in Section III-A, an arbitrary choice of Q_2 may lead to conservatism in the optimal κ achieved. To improve the value of κ, we propose an iterative method that makes use of Theorem 3.3 and updates Q_2:

We note that when M and K are identity and Q_2 is initialized as above, the columns of Q_2 are orthogonal both to each other and to $\mathbf{1}_n$.

Numerical Example: For a twenty node unweighted chain graph obeying the structure of Figure 1, we have:

$$\lambda_2(L) = 0.0246 \text{ and } \kappa = \frac{\lambda_{20}(L)}{\lambda_2(L)} = 161.6016.$$

We set $\epsilon = 10^{-2}$, and apply our method to reduce κ. For the first three experiments, (11) was solved with Q_2 set to

Algorithm 1 Iterative Updates for Q_2

1: $M \Leftarrow I$, $K \Leftarrow I$, $\mu > 0$.
2: **repeat**
3: Set Q_2 to be the matrix whose columns are the eigenvectors corresponding to the $n-1$ largest eigenvalues of $AKA^T - \underline{\lambda}_2 M$.
4: Solve (11) and update M, K.
5: **until** $|\kappa_i - \kappa_{i-1}| \leq \mu$
OR $\max(M) = \frac{1}{\epsilon}$ AND $\min(M) = \epsilon$
OR $\max(K) = \frac{1}{\epsilon}$ AND $\min(K) = \epsilon$.

be a matrix whose $n-1$ columns are orthogonal to $\mathbf{1}_n$. The lower eigenvalue bound was set to be $\underline{\lambda}_2 = \lambda_2(L)$. Solving for edges only, with nodes weighted to identity, produced no re-weighting of edges, and so κ was unchanged. In contrast, solving for nodes only, with edges weighted to identity, resulted in $\kappa = 123.5286$. Simultaneous optimization with both the nodes and edges as decision variables produced a marked improvement to $\kappa = 52.8616$. Allowing Q_2 to vary in accordance with Algorithm 1 described above resulted in $\kappa = 13.0499$. By setting $\epsilon = 10^{-3}$, we achieved $\kappa = 6.3021$.

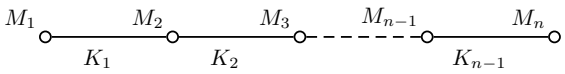

Fig. 1. A chain graph with n nodes.

B. Minimizing the Gap between λ_p and λ_{p+1}

We consider graphs with *clusters*, that is, groupings of densely connected nodes with sparse external links. The Laplacian of a graph with p clusters exhibits, in addition to the first eigenvalue at zero, $p-1$ additional eigenvalues close to zero. Such graphs exhibit a gap between the first p eigenvalues and the rest. Examples of systems with clustered structures have been observed in building sensor networks [21] and power systems [22], where distributed estimation algorithms are increasingly prevalent. The gap in the eigenvalues may be undesirable because it leads to a two-time-scale behavior in the convergence of these algorithms [23].

To obtain uniform convergence rates for nodes in different clusters, we maximize λ_2 while requiring $\lambda_{p+1} \leq \overline{\lambda}_{p+1}$, and in so doing, minimize the gap between $\lambda_p(L_g)$ and $\lambda_{p+1}(L_g)$. Additionally, we fix $\lambda_n(L_g) \leq \overline{\lambda}_n$, so that the rest of the spectrum of the weighted Laplacian does not deviate far from its original location. The problem is solved with a bisection to maximize κ on the interval $[\lambda_2(L), \overline{\lambda}_{p+1}]$. We impose upper and lower bounds on the entries of M and K, and introduce $Q_2 \in \mathbb{R}^{n\times(n-1)}$ and $U_{p+1} \in \mathbb{R}^{n\times(p+1)}$ defined according to Theorems 3.2 and 3.4, respectively. We now write the quasiconvex problem:

$$\begin{aligned} \underset{\kappa,M,K}{\text{maximize}} \quad & \kappa \\ \text{subject to} \quad & \overline{\lambda}_n M - AKA^T \succeq 0 \\ & Q_2^T(AKA^T - \kappa M)Q_2 \succeq 0 \\ & U_{p+1}^T(\overline{\lambda}_{p+1}M - AKA^T)U_{p+1} \succeq 0 \\ & \epsilon^{-1}I \succeq M \succeq \epsilon I,\ \epsilon^{-1}I \succeq K \succeq \epsilon I \\ & M,\ K \text{ diagonal.} \end{aligned} \tag{12}$$

We can realize significant improvements in reducing the gap between $\lambda_p(L_g)$ and $\lambda_{p+1}(L_g)$ by employing an iterative procedure that makes use of Theorems 3.3 and 3.5 as shown in Algorithm 2.

Algorithm 2 Iterative Updates for Q_2, U_{p+1}

1: $M \Leftarrow I$, $K \Leftarrow I$, $\mu > 0$.
2: **repeat**
3: Set Q_2 to be the matrix whose columns are the eigenvectors corresponding to the $n-1$ largest eigenvalues of $AKA^T - \underline{\lambda}_2 M$.
4: Set U_{p+1} to be the matrix whose columns are the eigenvectors corresponding to the $p+1$ smallest eigenvalues of $\overline{\lambda}_{p+1}M - AKA^T$.
5: Solve (12) and update M, K.
6: **until** $|\kappa_i - \kappa_{i-1}| \leq \mu$
OR $\max(M) = \frac{1}{\epsilon}$ AND $\min(M) = \epsilon$
OR $\max(K) = \frac{1}{\epsilon}$ AND $\min(K) = \epsilon$.

Numerical Example: Consider the eight node graph with two clusters in Figure 2. Such a graph, with identical weights, exhibits a significant gap between λ_2 and λ_3. The eigenvalues of the unweighted graph are:

$$\{0.0000,\ 0.3542,\ 4.0000,\ 4.0000,\\ 4.0000,\ 4.0000,\ 4.0000,\ 5.6458\},$$

with:

$$\frac{\lambda_3(L)}{\lambda_2(L)} = 11.2931.$$

Our goal is to reduce this gap by increasing the second eigenvalue while bounding the third and eighth eigenvalues from above.

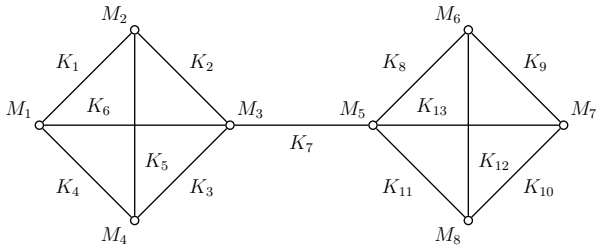

Fig. 2. Eight node graph with two clusters.

We employ Algorithm 2, iteratively updating both Q_2 and U_3 while requiring $\lambda_3(L_g) \leq 4.0000$ and $\lambda_n(L_g) \leq 5.6458$

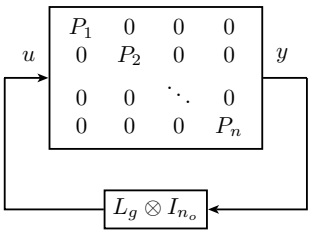

Fig. 3. Block diagram of the multi-agent system (13-14).

and setting $\epsilon = 10^{-2}$. We find the optimal value $\lambda_2(L_g) = 4.0000$, with $\frac{\lambda_3(L_g)}{\lambda_2(L_g)} = 1$.

V. Application to Multi-Agent Systems

We now apply the results of Section IV-A to multi-agent systems whose feedback structure is described by a graph Laplacian. Each of the n subsystems possesses identical dynamics:

$$P_i : \left\{ \begin{array}{rcl} \dot{x}_i &=& Fx_i + Gu_i \\ y_i &=& Hx_i \end{array} \right. \tag{13}$$

and is controlled according to the feedback law:

$$u_i = -M_i^{-1} \sum_{j \in \mathcal{N}_i} K_j (y_i - y_j), \tag{14}$$

where $F \in \mathbb{R}^{n_s \times n_s}$, $G \in \mathbb{R}^{n_s \times n_o}$, and $H \in \mathbb{R}^{n_o \times n_s}$, with n_s and n_o the dimension of the state space and input and output, respectively. $\mathcal{N}_i$ denotes the neighbors of agent i, that is, the other agents whom agent i senses. We assume that the individual plants are stable or can be stabilized by local state feedback (see the numerical example below). Therefore, we assume that F is Hurwitz. M_i and K_j denote entries i and j of the diagonal node and edge weighting matrices M and K, respectively. The block diagram of the system is shown in Figure 3, with each subsystem P_i having input given by (14). We let $x = [x_1^T, \ldots, x_n^T]^T$, and rewrite (13) and (14) as:

$$\dot{x} = [I_n \otimes F - L_g \otimes (GH)]x. \tag{15}$$

As a consequence of the identical dynamics of each subsystem, the system can be decoupled into n identical subsystems by a change of coordinates using the basis of eigenvectors of L_g [3]. Let U be the orthogonal change-of-coordinates matrix that diagonalizes L_g and let D be the diagonal matrix of eigenvalues of L_g. Then $D = U^{-1} L_g U$. Now let $V = U \otimes I$, and let $\tilde{x} = V^{-1}x$. In the new coordinates, the dynamics are given by:

$$\dot{\tilde{x}} = [I_n \otimes F - D \otimes (GH)]\tilde{x}, \tag{16}$$

and, thus, the eigenvalues are determined from the characteristic polynomials of:

$$F - \lambda_i(L_g)GH,\ i = 1, \ldots, n. \tag{17}$$

This means that the multi-agent system can be analyzed as n decoupled feedback systems with constant gain $\lambda_i(L_g)$, $i = 1, \ldots, n$. In particular, larger Laplacian eigenvalues imply higher gains for these decoupled systems, which is often undesirable. For example, if the transfer function $H(sI - F)^{-1}G$ has non-minimum phase zeros or relative degree higher than two, high gain will result in right half plane poles, rendering the multi-agent system unstable. The largest eigenvalue minimization method of Section IV-A can mitigate this instability by finding a node and edge weighting such that the spectrum of L_g spectrum falls within a range specified by design requirements.

Numerical Example: We consider formation control for four planar vertical takeoff and landing, or PVTOL, aircraft, as described in [24]. We model the state of the aircraft by its lateral position, x, vertical position y, and its roll, θ. The equations of motion, in input-output linearized form, are given by the following:

$$\begin{array}{rcl} \ddot{x} &=& u_1 \\ \ddot{y} &=& u_2 \\ \ddot{\theta} &=& \frac{1}{\epsilon}(\sin\theta + \cos\theta u_1 + \sin\theta u_2). \end{array} \tag{18}$$

The zero dynamics of the system are unstable:

$$\ddot{\theta} = \frac{1}{\epsilon} \sin\theta, \tag{19}$$

and the system is non-minimum phase.

We assume that the aircraft are in hover operation and are stabilized vertically, so we discard y and v_2, the vertical thrust input. We set $\epsilon = 0.1$ and see that the linearized dynamics around $x = 0$, $\theta = 0$ are:

$$\tilde{F} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 10 & 0 \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 10 \end{bmatrix} \quad H = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T.$$

The input to each aircraft is dictated by the input term of (13), with the graph structure of a four node chain. We choose a state feedback:

$$J = [\ 0 \quad -90.6157 \quad 42.1472 \quad 13.2155\ ],$$

which renders $F = \tilde{F} - GJ$ Hurwitz. To achieve a reasonable response time and to maintain stability, we wish to contain the eigenvalues of the Laplacian in the interval $[50, 125]$. In particular, the upper bound of this interval guarantees a damping ratio greater than 0.6. For the unweighted Laplacian, we have:

$$\kappa = \frac{\lambda_n(L)}{\lambda_2(L)} \leq 5,$$

which means that scaling the Laplacian by a constant $\alpha = \frac{50}{\lambda_2(L)}$ to meet the lower eigenvalue constraint $\lambda_2(\alpha L) \geq 50$ will violate the upper eigenvalue constraint $\lambda_n(\alpha L) \leq 125$ and lead to instability as illustrated in Figure 4. In contrast, applying the node and edge weights found by applying Algorithm 1 results in an improvement to $\kappa = 1.0202$. We show simulation results with the new weights in Figure 5.

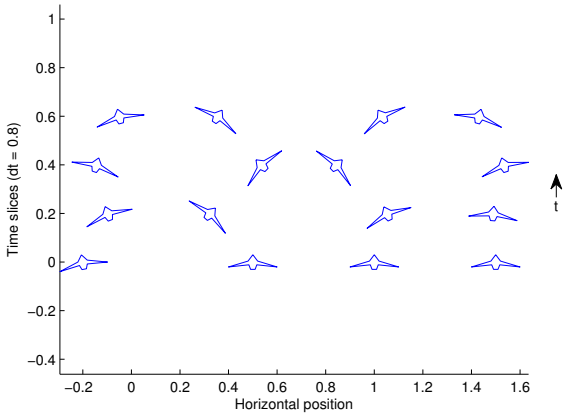

Fig. 4. PVTOL formation of four aircraft with unweighted, scaled graph. Each row represents a snapshot in time, indicating unstable behavior. Both the maximum roll angle and the amplitude of deviation from the desired relative position of the aircraft increase in time.

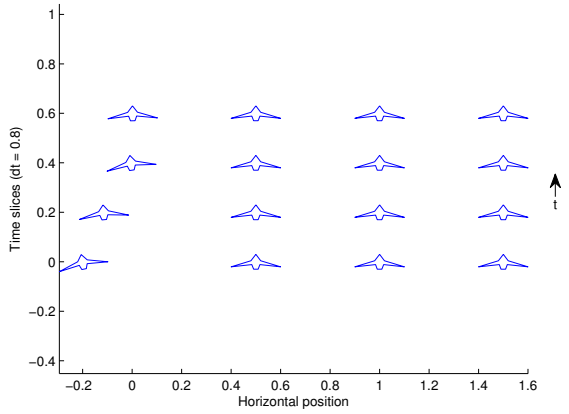

Fig. 5. PVTOL formation of four aircraft with weighted graph. Each row represents a snapshot in time, indicating convergence to formation. Both the maximum roll angle and the amplitude of deviation from the desired relative position of the aircraft decrease in time.

VI. Conclusion

The graph Laplacian is an indispensable tool for assessing the dynamics of a multi-agent system. In this paper, we have presented a novel approach to impose bounds on the Laplacian spectrum. We have shown how node and edge weights can be adjusted using convex optimization to impose individual constraints on several eigenvalues simultaneously. In future work, we will quantitatively characterize any optimality gaps and convergence properties for Algorithm 2. We are also examining first order methods [25][26] that will allow our framework to accommodate systems with thousands of agents.

References

[1] F. Chung, "Spectral graph theory," *American Mathematical Society*, 1997.

[2] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[3] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[5] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[7] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

[8] H. Bai and M. Arcak, "Instability mechanisms in cooperative control," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 258–263, 2010.

[9] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Society for Industrial Mathematics, 1994.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[11] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Review*, vol. 50, no. 1, p. 37, 2008.

[12] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM Review*, vol. 46, no. 4, pp. 667–689, 2004.

[13] J. Sun, S. Boyd, L. Xiao, and P. Diaconis, "The fastest mixing Markov process on a graph and a connection to a maximum variance unfolding problem," *SIAM Review*, vol. 48, no. 4, p. 681, 2006.

[14] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, p. 117, 2006.

[15] Y. Shafi, M. Arcak, and L. El Ghaoui, "Designing node and edge weights of a graph to meet Laplacian eigenvalue constraints," in *Prof. Allerton Conference*, 2010.

[16] R. Horn and C. Johnson, *Matrix analysis*. Cambridge Univ Pr, 1990.

[17] F. Gantmacher, *The theory of matrices*. Chelsea Pub Co, 2000.

[18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Oct. 2010.

[19] ——, "Graph implementations for nonsmooth convex programs," *Recent advances in learning and control*, pp. 95–110, 2008.

[20] K. Toh, M. Todd, and R. Tutuncu, "SDPT3a Matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, no. 12, pp. 545–581, 1999.

[21] J. Kim, M. West, E. Scholte, and S. Narayanan, "Multiscale consensus for decentralized estimation and its application to building systems," in *American Control Conference, 2008*. IEEE, 2008, pp. 888–893.

[22] J. Chow, Ed., *Time-Scale Modeling of Dynamic Networks with Applications to Power Systems*. Berlin Heidelberg: Springer-Verlag, 1982.

[23] E. Biyik and M. Arcak, "Area aggregation and time-scale modeling for sparse nonlinear networks," *Systems & Control Letters*, vol. 57, no. 2, pp. 142–149, 2008.

[24] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer Verlag, 1999.

[25] Z. Wen, D. Goldfarb, and W. Yin, "Alternating direction augmented Lagrangian methods for semidefinite programming," *Mathematical Programming Computation*, pp. 1–28, 2010.

[26] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, "Row by row methods for semidefinite programming," *Technical Report, Department of IEOR, Columbia University*, 2009.