Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

On Time-Scale Designs for Networks

Sandip Roy, Yan Wan, and Ali Saberi

Abstract— **We motivate the problem of designing a subset of the edge weights in a graph, to shape the spectrum of an associated linear time-invariant dynamics. We address a canonical design problem of this form by applying time-scale assignment methods, and give graph-theoretic characterizations of the designed dynamics.**

I. Introduction

Controller design for the purpose of time-scale assignment is a cornerstone of classical and modern control theory. In such controllers, high and/or low gains of various scales are used to assign the eigenvalues of a linear time-invariant (LTI) plant along one or more asymptotic time scales, e.g. [1], [2]. Time-scale assignment has proved critical for a family of stabilization and performance-design tasks. In particular, multiple time-scales are fundamentally needed for disturbance rejection (e.g. [3]), and further permit systematic solution of such varied problems as stabilization/regulation under actuator saturation (e.g. [4]), loop transfer recovery (e.g. [2]), and decentralized controller design [5], among others. The ability to assign eigenvalues along desired time-scales is fundamentally related to the *linear-system structure*, i.e. to the zero dynamics and infinite-zero structure of the plant. As designs for large-scale systems and networks are increasingly needed, however, it is becoming more and more important that time-scale assignment capabilities be related to the *topological* (graph) structure of the system. To clarify this connection, the zero- and infinite-zero- structure—and hence the time-scale design properties—must be characterized in terms of the topological structure.

In a complementary direction, the common presence of multiple time-scale dynamics in existing large-scale infratructure networks has been explained, and the time-scale structure *has* been related to the topological structure of the network (e.g., [6]). This characterization—which originated in the electric power systems community under the heading of **slow coherency** and was further generalized through the definition of **synchrony** [6]—is based on the premise that large-scale networks naturally have groups of components that are strongly connected to each other but only weakly tied to the remainder of the network. The special topological structure yields 1) slow dynamics that are global but coherent or synchronic within each tightly-connected group, and 2) fast dynamics that are localized to individual groups. This recognition of the typical time-scale structure of networks is valuable for a family of infrastructure-network analyses, including for model reduction and partitioning (e.g., [6]). However, the graph structure-based time-scale characterizations are only for existing networks, and the idea of *designing* desirable time scales by exploiting the graph structure has not been addressed. Such design is of significant interest, because it can permit shaping of the network dynamics, including specifically the modification of existing coherency structures.

The purpose of this this work is to marry the efforts on time-scale assignment with the graph-structural characterization of time-scales in large-scale network analysis. That is, we motivate the problem of *designing time-scales in large scale systems by exploiting their topological structure*, and in turn initiate research in this direction by addressing a canonical design problem of wide interest. Precisely, we identify several controller design and graph-edge design problems in networks, for which time-scale designs that exploit the network's topology are needed. The problems that we identify originate from such diverse fields as virus-spreading control, drug design, traffic flow management, and sensor networking. We then fully address the time-scale design for an example problem motivated by these applications, namely that of designing some edge weights in a graph (while others remain fixed) to shape a dynamics defined by the associated **Laplacian** matrix (see [7] for background on the Laplacian matrix and its spectrum). Specifically, viewing this *partial graph design* problem as a (decentralized) controller design problem, we characterize the infinite-zero structure and finite-zero dynamics of the plant in terms of the topologies of the fixed and designable graph edges. In turn, we propose a high-gain methodology for the partial graph design, and characterize the spectrum upon design in terms of the graph topologies. We thus tie the performance of time scale-based designs to the topological structure, and (in a complementary direction) characterize the network dynamics over a range of edge-weight values.

In that we are obtaining designs for networks that exploit their graph structure, our efforts here contribute to the nascent research on high-performance network design [5], [8]–[10], [12]. These studies are focused on designing network controllers or connections (edges) to optimize dynamic measures, often using optimization machinery along with algebraic graph-theory notions. These design problems are complex, and only the simplest cases have been addressed—e.g., designing all the transition probabilities in a Markov chain to achieve fast mixing, or selecting static controllers for a network of autonomous agents with single-integrator dynamics [8], [9]. Our efforts here expose that time-scale designs can be used for a much wider class of design prob-

All authors are with the Electrical Engineering Department, Washington State University, and can be reached at {sroy, ywan saberi}@eecs.wsu.edu.
This work was supported by NSF Grants ECS-0528882 and ECS-0725589, and by NASA Grant NNA06CN26A. We thank Dr. B. Lesieutre for a discussion on the subject.

lems, in particular ones where the network has a complex existing structure and only some local features (whether edge properties or local controllers) can be altered. For these partial design problems, the dynamics from the point of the designable features exhibit a rich structure that is deeply tied to the existing network topology; a graph-based approach to time-scale design is critical for shaping these dynamics.

The remainder of the paper is organized as follows. In Section II, we motivate partial graph design problems in several applications. In Section III, we address a canonical partial graph design problem, and also present an example illustrating our design. We omit all proofs in the interest of space, please see the extended document [16] for them.

II. Motivation

We motivate the partial graph design problem, and hence the time-scale assignment methodology that exploits topological structure, from several application domains. Because of the limited space, we only discuss the design in vechile/sensor networking in detail. Please find the discussiond on epidemic spread control, multi-target drug design, and power network design in [16].

Recently, the development of distributed algorithms/controllers for autonomous vehicle coordination and sensor networking applications has been of wide interest in the controls community (e.g., [11]). Such controller/algorithms have been developed for many tasks (including formation and distributed partitioning), but these various tools have in common that they permit highly-limited agents to coordinate by exploiting network interactions. Although many network tasks have been studied, however, design of *high-performance* controllers/algorithms (that achieve fast settling, and robustness to disturbances and variations) is in its early stages (e.g. [8], [9]). These few efforts have pursued designing the whole network topology, or local controllers at all network nodes, to optimize a performance measure (e.g., a settling rate or condition number) [9], [10], [12]. Building on these, we have demonstrated that pole placement can be achieved using multiple-delay controls at all network nodes [5].

In many cases, only partial design of the network interactions and controllers is possible. For instance, only a few nodes in a large-scale mobile sensor network may be amenable to modification, due to limitations in resources or access. Similarly, sensor networks that are operated by multiple players perhaps only can be updated in parts, only newly-added sensors/vehicles may be amenable to modification, or only certain communication links may admit higher bandwidth/fidelity. In these cases, design of a subset of the network edges (specifically, protocol strengths or weights) and nodes (specifically, controller gains) is of critical interest. This is precisely the partial graph design problem.

In a complementary direction, we often need to characterize the dynamics of autonomous-vehicle teams or sensor-network algorithms upon perturbation of network parameters, due to e.g. communication failures or environmental variation. Such characterizations also require us to study network dynamics as edge weights or controller gains are varied, and so partial graph design informs the analysis.

A wide variety of agent, network, and controller models are used in sensor networking and autonomous vehicle control applications, and so a range of partial graph design problems can be posed. Here, let us abstractly introduce only one canonical design problem, which for instance is representative of consensus-algorithm and velocity-coordination design, e.g. [9], [11]. Specifically, let us consider the dynamics $\dot{\mathbf{x}} = -L\mathbf{x}$, where $\mathbf{x}$ represents the agents' states (e.g., velocities, opinions), and L is the Laplacian matrix associated with a weighted graph Γ that has vertex set V and edge set E, and weight $k_{ij} > 0$ for each $\{i, j\} \in E$. Let the edge set E consist of a subset E_f with edges having fixed weights and a subset $E_d = E - E_f$ with edges whose weights can be designed. The design problem of interest is to select the weights k_{ij} for $\{i, j\} \in E_d$, so as to shape the dynamics $\dot{\mathbf{x}} = -L\mathbf{x}$ (in particular, to decrease the settling time of the dynamic response while limiting the impact of initial-condition- and external- disturbances).

III. Partial Graph Design: Canonical Example

The various design problems introduced in Section II and [16] have a common aim: we seek to design the strengths of *some* interconnections in a network, or else decentralized controllers at some network nodes, to shape the network's dynamics. Furthermore, we aim for designs and design characterizations phrased in terms of the network's structure.

To this end, we here pursue a canonical partial network design problem, specifically the problem of designing a subset of the edges' weights in a graph to shape the dynamics defined from an associated Laplacian matrix. We introduce the design problem in Section IIIA and reformulate it as a decentralized controller design problem in Section IIIB. From the reformulation, we take two steps: 1) we relate the linear system structure (finite- and infinite-zero structure) of the open-loop plant to the network's topology, in particular a *fixed-edge graph* (i.e., comprising edges that cannot be designed) and a *designable-edge graph* (comprising edges whose weights can be selected) (Section IIIC); 2) by using the time-scale-based design methodology [1], [2], we address the partial network design problem from a controller design viewpoint (Section IIID) and so characterize the closed-loop spectrum. In this way, we both obtain and characterize designs in terms of the network's topology. Finally, we give an example (Section IIIE).

A. Formulation

We focus on designing the weights of a subset of the edges in a graph, to shape the spectrum of an associated Laplacian matrix and hence to shape dynamics defined thereof. This design problem for Laplacians is directly applicable to two of the applications in Section II, namely the sensor networking and electric power applications. We stress, however, that the methodologies for this particular design problem naturally can be adapted to the various other controller and network-interconnection design problems posed in the introduction *including one with asymmetric topologies*.

Precisely, we consider a weighted and undirected graph Γ with n vertices, labeled $1,\dots,n$. We specify the edges in the graph through two disjoint sets each containing pairs of distinct vertices, which we term the **fixed edge set** E_f and the **designable edge set** E_d. Specifically, for each pair of vertices $\{i,j\}\in E_f$, the graph Γ has an edge between vertex i and j with fixed weight $k_{ij}>0$. Meanwhile, for each pair $\{i,j\}\in E_d$, the graph has an edge between vertex i and vertex j with weight k_{ij} that can be set to a desired nonnegative value. For pairs $\{i,j\}$ that are neither in E_d or E_f, we shall say that there is not an edge between vertex i and vertex j, and for convenience we set the weight k_{ij} to 0. We also find it convenient to label and order the edges in E_d with the positive integers $1,\dots,|E_d|$, and refer to the weight of the edge $m\in\{1,\dots,|E_d|\}$ as k_m.

We aim to design the edge weights k_{ij} for $\{i,j\}\in E_d$, so as to shape a dynamics defined from the weighted Laplacian matrix associated with the graph Γ (e.g.! [7]). Let us recall that the Laplacian matrix associated with the graph Γ, which we denote as $L(\Gamma)$, is defined as follows: $[L(\Gamma)]$ is an $n\times n$ matrix with entries given by $[L(\Gamma)]_{ij}=[L(\Gamma)]_{ji}=-k_{ij}$ for all $i\neq j$, and $[L(\Gamma)]_{ii}=-\sum_{j\neq i}[L(\Gamma)]_{ij}$ for all i. Our goal is to design the edge weights $k_{ij}\in E_d$, to shape the spectrum of $L(\Gamma)$ (i.e., to assign its eigenvalues and eigenvectors), or equivalently to shape the dynamics of such differential equations as $\dot{\mathbf{x}}=L\mathbf{x}$, $\dot{\mathbf{x}}=-L\mathbf{x}$, or $\ddot{\mathbf{x}}=-L\mathbf{x}$.

We refer to the above design problem in its entirety as the **partial graph design problem**. For convenience, we refer to the graph Γ in the case where the designable edge weights are set to zero as the **fixed-edge graph**, and use the notation Γ_F for it. We use the notation $L(\Gamma_F)$ for the corresponding Laplacian. We also form a **designable-edge graph** Γ_D by removing the fixed edges from Γ; we define the Laplacian matrix $L(\Gamma_D)$ for the designable graph in the standard way.

B. Reformulation as a Decentralized Controller Design

We aim to set the designable edge weights in graph Γ to shape the dynamics of $\dot{\mathbf{x}}=L(\Gamma)\mathbf{x}$, or in other words assign the spectrum of $L(\Gamma)$. The task of shaping the dynamics can be reformulated as a linear static decentralized controller design problem. Through designing the controller's gains, we in turn are able to assign the spectrum of $L(\Gamma)$.

To present the reformulation, we find it convenient to use the notation $\mathbf{q}_i^j$ for the n-component vector with ith entry equal to 1, jth entry equal to -1, and remaining entries null. In this notation, the Laplacian $L(\Gamma)$ can be rewritten as $L(\Gamma)=\sum_{\{i,j\}\in E}k_{ij}\mathbf{q}_i^j{\mathbf{q}_i^j}^T=\sum_{\{i,j\}\in E_f}k_{ij}\mathbf{q}_i^j{\mathbf{q}_i^j}^T+\sum_{\{i,j\}\in E_d}k_{ij}\mathbf{q}_i^j{\mathbf{q}_i^j}^T$.

To clarify that the design of $L(\Gamma)$ is a decentralized controller design problem, let us define the following matrices:

- We let $A=\left[\sum_{\{i,j\}\in E_f}k_{ij}\mathbf{q}_i^j{\mathbf{q}_i^j}^T\right]$.
- For each edge $m=1,\dots,|E_d|$ in the designable edge set, let B_m equal q_i^j, where $\{i,j\}$ are the two ends of the edge. Also, we let $C_m=B_m^T$.
- Recall that we denote the weight of edge m in the designable-edge graph by k_m.

In this notation, we immediately recover the following expression for $L(\Gamma)$: $L(\Gamma)=A+\sum_{m=1}^{|E_d|}B_mk_mC_m$. Noting that the weights k_m are the parameters in $L(\Gamma)$ amenable to design, we see that the partial graph design problem is the following decentralized controller design problem: a decentralized LTI plant with state matrix A has $|E_d|$ channels, where channel m has observation matrix C_m and actuation matrix B_m; static nonnegative linear feedback gains k_m for all m such that $m\in\{1,...,|E_d|\}$ must be developed at the $|E_d|$ channels, to shape the dynamics $\dot{\mathbf{x}}=L(\Gamma)\mathbf{x}$, or equivalently the spectrum of $L(\Gamma)$.

For convenience, let us stack each channel's observations and inputs into matrices, as follows: $B\triangleq\begin{bmatrix}B_1 & \dots & B_{|E_d|}\end{bmatrix}$, and $C\triangleq\begin{bmatrix}C_1 & \dots & C_{|E_d|}\end{bmatrix}^T$. In this notation, the decentralized system's closed-loop dynamics can be written as $\dot{\mathbf{x}}=A\mathbf{x}+BKC\mathbf{x}$, where K is an $|E_d|\times|E_d|$ diagonal matrix whose diagonal entries are the designable edge weights. Thus, we see that the partial graph design problem can be viewed as an $|E_d|$ channel static decentralized controller design for the plant (C,A,B), for the purpose of spectrum assignment. In the remainder of this section, we obtain and characterize solutions to this design problem, that are based on the topology of the fixed- and designable- edge graphs.

C. Topological Characterization of the Plant Dynamics

Time-scale assignment for LTI plants requires characterizating the *linear-system-structure*, i.e. the infinite-zero- and finite-zero dynamics of the plant. The **special coordinate basis** (SCB) for linear systems provides a representation of the linear system structure that allows time-scale design [13]. Thus, here we obtain the SCB for the formulated plant model, as a step toward time-scale assignment through partial graph design. We note that the SCB was developed for centralized control; however, this work and also [5] indicate that the SCB permits decentralized controller design also.

The special structure of the partial graph design problem permits us to characterize the linear system structure of the plant (C,A,B) in terms of the graph topology. We begin with a preliminary remark on the plant's open-loop poles:

Remark: The open-loop poles of the plant (C,A,B) are the eigenvalues of the matrix $L(\Gamma_F)$.

Next, we present several results that together specify the finite- and infinite-zero dynamics of the plant (C,A,B) in terms of the fixed- and designable- edge graph topologies. For convenience, we do so (Theorems 1$-$3) in the case where the designable graph Γ_D is is a **z-forest**, i.e. a collection of z **trees** or connected acyclic graphs. From the perspective of obtaining the linear system structure, we can limit ourselves to the case where the designable graph is acyclic WLOG, since cyclic designable graphs yield redundant observation and input, i.e. the matrices B and C are not full rank. In particular, one can always view the control as using a subset of observations and inputs and hence define the finite- and infinite- zero dynamics thereof, while the redundant observations and inputs are simply considered unused. Thus, the results that we obtain for the linear-system structure for the z-forest case trivially translate to the general case, and so we focus on the z-forest case for notational simplicity. It is worth making one further observation, however: redundancy in B and C is of no value to design in the centralized setting, hence the centralized infinite-zero and finite-zero structure decomposition always ignores this case. Interestingly, however, the use of non-tree designs fundamentally affects other aspects of performance in the decentralized setting. We thus give a brief discussion of the cyclic designable graph design in Section IIID. As an aside, we note that acyclic designable graphs appear in many applications, e.g. in design of a single interaction (edge), or in designing interconnections between newly added nodes and existing ones.

In characterizing the finite- and infinite-zero dynamics, let us first specify the dimensions of each, and so clarify that the plant has a **uniform-rank** structure (see e.g. [14]):

Theorem 1: When Γ_D is a z-forest with $|E_d|$ edges, the plant (C, A, B) is square-invertible and uniform-rank-1, with finite- invariant zero dynamics of dimension $n - |E_d|$.

The uniform-rank-1 structure of the plant immediately permits us, through simple transformation of the standard representation for uniform-rank system, to phrase the dynamics of the plant (C, A, B) as follows:

$$\begin{aligned}\dot{\mathbf{y}} &= P_1\mathbf{y} + C_a\mathbf{x}_a + Q\mathbf{u} \\ \dot{\mathbf{x}}_a &= A_a\mathbf{x}_a + B_a\mathbf{y},\end{aligned} \tag{1}$$

where $\mathbf{x}_a \in R^{n-m}$ represents the state of the plant's finite-invariant zero dynamics, and the matrices P_1, C_a, Q, A_a, and B_a are obtained through the state transformation. Next, we focus on characterizing the parameters of the SCB representation in terms of the fixed- and designable- edge graphs' topologies, to permit high-gain design and design characterization in terms of the graph topology.

Our characterization of the infinite-zero and finite-zero dynamics is in two steps: first (Theorem 2), we define a state vector for the finite-zero dynamics, and so specify the finite- and infinite-zero dynamics formally in terms of the plant model (C, A, B) (but in a form that facilitates connection with the graph topology). Next (Theorem 3), we give an explicit graph-theoretic construction of the state matrix associated with the finite-zero dynamics. Before presenting these results, we require some further notations.

—We call x_i the state variable **associated** with vertex i.

—We find it convenient to partition the vertices based on the graph Γ_D. In particular, we partition the vertex set in such a way that two vertices are in the same partition if and only if there is a path between them in Γ_D. Notice that the groups of vertices that form connected subgraphs in Γ_D, as well as the remaining isolated vertices, are the partitions. In total, there are $n - |E_d|$ partitions, which we label $S_1, \ldots, S_{n-|E_d|}$.

—We define the **superstate** $\overline{x}_i$ associated with each partition $i = 1, \ldots, n - |E_d|$, as $\overline{x}_i = \frac{1}{|S_i|}\sum_{j \in S_i} x_i$. We notice that a vector containing the super-states can be computed as a linear combination of the state vector $\mathbf{x}$, say as $\widehat{C}\mathbf{x}$.

We are now ready to specify a state for the zero dynamics, and hence obtain the SCB representation formally in terms of the plant model.

Theorem 2: The superstates $\overline{x}_i$, $i = 1, \ldots, n - |E_d|$, together form a state for the zero dynamics of the plant. In these coordinates, the SCB representation of the plant is

$$\begin{aligned}\dot{\mathbf{y}} &= CA\begin{bmatrix} C \\ \widehat{C}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{y} \\ \mathbf{x}_a\end{bmatrix} + CB\mathbf{u} \\ \dot{\mathbf{x}}_a &= \widehat{C}A\begin{bmatrix} C \\ \widehat{C}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{y} \\ \mathbf{x}_a\end{bmatrix}.\end{aligned} \tag{2}$$

While Theorem 2 formally specifies the SCB of the plant, the representation is not explicitly connected to the fixed- and designable- graphs' topologies. In fact, all the parameter matrices in the SCB representation can be described explicitly in terms of the graph topologies. In the interest of space, we only characterize the state matrix of the zero dynamics, which is of critical importance in the high-gain design, in this way. We first require a bit further notation: Consider two distinct partitions S_i and S_j, where $i, j \in 1, \ldots, n - |E_d|$. We use the notation $\overline{k}(S_i, S_j)$ for the sum of the (fixed-graph) edge weights between partitions i and j, i.e. $\overline{k}(S_i, S_j) = \sum_{l \in S_i, m \in S_j} k_{lm}$. We refer to $\overline{k}(S_i, S_j)$ as the **aggregate weight** between partitions i and j.

We are now ready to present the structural result:

Theorem 3: Consider the partial edge design problem when the designable-edge graph is a z-forest, and consider the state matrix of the zero dynamics A_a in (1). The entry at row i and column j of A_a is given by $\frac{\overline{k}(S_i,S_j)}{|S_i|}$, for $i \neq j$. The diagonal entries are the negative of the sum of the off-diagonal entries, i.e. they make the row sums zero.

Let us make a couple observations about the structural result given in Theorem 3. First, we note that the state matrix of the zero dynamics is a Laplacian matrix with each row i inversely scaled by the number of vertices in partition i. More specifically, the zero dynamics is defined by a Laplacian matrix of a graph with the aggregate weights specifying the edge values, together with a diagonal scaling matrix. Precisely, let us define the **zero graph**.Γ_Z as a weighted and undirected graph with $n - |E_d|$ vertices, with the edge between vertex i and vertex j having weight $\overline{k}(S_i, S_j)$. We refer to the Laplacian of the graph (defined in the standard way) as $L(\Gamma_Z)$. Further, let us define the **size-scaling matrix** D as an $|E_d| \times |E_d|$ diagonal matrix with entries given by $|S_i|$. In this notation, the state matrix of the zero dynamics is given by $D^{-1}L(\Gamma_Z)$. From this expression, we see that the plant (C, A, B) has at least one zero at the origin, with the number at the origin given by the number of components in the graph Γ_Z. It is easy to check that the zeros at the origin are both *input- and output- decoupling zeros*. Meanwhile, the remaining zeros are strictly positive, and are transmission zeros. Since the zeros are the eigenvalues of a scaled Laplacian matrix, algebraic graph theory tools can be brought to bear to characterize the zeros (e.g. [7]).

D. Time-Scale Design

A high-gain controller architecture permits systematic assignment of the closed-loop poles of an LTI plant along asymptotic time scales, see the literature on **asymptotic time-scale and eigenstructure assignment**, or ATEA, design [2]. Specifically, given a plant's infinite-zero structure (which is clarified by the SCB), one can specify a family of high-gain controllers that allow placement of certain eigenvalues at one or more desirable *fast* time scales, while the remaining *slow* eigenvalues approach the finite invariant zeros of the plant. These multiple time-scale designs are widely used, including for design of stabilizating controllers [1], almost-disturbance-decoupling [3], and (as we have recently clarified) decentralized controller design [5]. As a further refinement, high-and-low-gain methods can be used for e.g. stabilization under saturation [4].

High-gain and high-and-low-gain methods are apt for the partial graph design problem. Specifically, in the various applications, large or small edge weights—which correspond to high- and low- gains in the controller design reformulation—are often naturally assigned: e.g., algorithm weights can be set to desired large or small values, or the rates of chemical reactions can be drastically changed in the context of drug design. For the complementary analysis tasks (e.g., characterization of settling rates over a range of uncertain weights), the design methods are also valuable because they specify the dynamics of the network over the range of possible parameter values and identify the extremes.

Here, we give a first set of results concerning high-gain approaches for the partial graph design problem[1]. Our focus here is on designing the initial-condition response of the plant and in particular the closed-loop spectrum, and also characterizing these designs in terms of the graph topology. We present our results in two steps, first specifying the high-gain design and its spectrum, and second discussing the spectrum over the range of possible edge weights by using the results from the design. Finally, we mention several generalizations and connections.

Let us begin by specifying and characterizing the high-gain design. Because of the special uniform-rank-1 structure of the equivalent controller design problem (as indicated in the SCB representation of the plant), we immediately recover from the ATEA design literature that identically scaling up all decentralized controller gains (equivalently, all designable edge weights) achieves a two-time-scale design. Specifically, we apply the following parameterized design: for each edge $\{i, j\} \in E_d$, we choose the edge weight as $k_{ij} = \alpha\widehat{k_{ij}}$ where each **nominal edge weight** $\widehat{k_{ij}}$ can be any fixed positive value, and the parameter α ($0 < \alpha < \infty$) provides an identical scaling to each weight. We use the notation $K(\alpha)$ for a family of edge designs of this form, and call these a **high-gain edge design**.

The following theorem specifies the two-time-scale structure resulting from use of a high-gain edge design:

Theorem 4: The spectrum of $L(\Gamma)$ upon application of a high-gain edge design $K(\alpha)$ with arbitrary nominal edge weights is as follows: for $\alpha \to \infty$, $L(\Gamma)$ has 1) $|E_d|$ eigenvalues that approach $+\infty$, and in particular are within $\mathcal{O}(1)$ of the non-zero eigenvalues of $L(\Gamma_D)$; and 2) $n-|E_d|$ eigenvalues that approach (i.e., are within $\mathcal{O}(\frac{1}{\alpha})$ of) the $n-|E_d|$ eigenvalues of $D^{-1}L(\Gamma_Z)$.

This theorem clarifies that *any* high-gain edge design drives $|E_d|$ (fast) eigenvalues of $L(\Gamma)$ arbitrarily far right in the complex plane, while moving the other (slow) eigenvalues toward those of the scaled zero graph (which are the in fact the zeros of the plant model (C, A, B)). The result follows immediately from consideration of the ATEA design [2] together with the SCB representation from Section IIIC.

When a high-gain edge-design is used, we can also characterize the eigenvectors of $L(\Gamma)$. Briefly, we find the following, for sufficiently large α. 1) The eigenvectors associated with the slow eigenvalues have (approximately) identical entries corresponding to vertices in the same partition; these entries are matched with the entries of the eigenvector of $D^{-1}L(\Gamma_Z)$. 2) The eigenvectors associated with the fast eigenvalues are each concentrated in the vertices corresponding to a single connected subgraph in Γ_D.

We stress that the above characterizations of the graph design hold for *any* high-gain edge design, i.e. regardless of the choice of the nominal edge weights. Let us briefly discuss how one can choose among possible nominal designs. In doing so, first let us note a fundamental difference between centralized and decentralized high-gain feedback: a centralized static output feedback can be used to place the $|E_d|$ fast eigenvalues at arbitrary locations, while in the decentralized setting the fast eigenvalues (i.e., the eigenvalues of $L(\Gamma_D)$) cannot be assigned arbitrarily. In fact, the problem of assigning the eigenvalues of $L(\Gamma_D)$ is that of designing *all* the edge weights in a specified graph to place the eigenvalues of the associated Laplacian at desirable locations, see the previous works [8], [15] for numerical/analytical solutions to this problem. It is worth noting that our capability for assigning the fast eigenvalues is highly dependent on whether or not Γ_D is a tree, demonstrating that the (decentralized) partial graph design problem depends intricately on the structure of Γ_D even though the linear system structure does not.

Next, we note that the high-gain design also provides insight into the Laplacian's spectrum, over a range of possible designable edge weights. Such insight is valuable for the complementary task of evaluating a network's dynamics, when some of the interconnection strengths are subject to variation. To this end, we show in the following theorem that the eigenvalues of $L(\Gamma)$ are bounded by those of the fixed-edge graph and zero-graph Laplacians.

Theorem 5: Consider a partial graph design problem. Denote the eigenvalues of $L(\Gamma_F)$ as $0 = \widehat{\lambda_0} \leq \widehat{\lambda_1} \leq \widehat{\ldots} \leq \widehat{\lambda_{n-1}}$, denote the eigenvalues of $D^{-1}L(\Gamma_Z)$ as $0 = \overline{\lambda_0} \leq \overline{\lambda_1} \leq \ldots \leq \overline{\lambda_{|E_d|-1}}$, and denote the eigenvalues of $L(\Gamma)$ by $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$. For any assignment of nonnegative edge weights in the designable graph, we have $\lambda_i \geq \widehat{\lambda_i}$, $i = 0, \ldots, n-1$ and $\lambda_i \leq \overline{\lambda_i}$, $i = 0, 1, \ldots, |E_d|-1$.
In other words, the ith eigenvalue of the graph Laplacian is between the eigenvalues of the fixed-edge graph's Laplacain and the scaled zero graph's Laplacian. The result follows

[1]We note that the results in this section hold for arbitrary designable-edge graphs, not only forests.

from Theorem 4 along with the property that a Laplacian's eigenvalues increase monotonically with the edge weights.

We have thus developed and characterized the performance of high-gain partial-graph designs. The fundamental contribution of this design methodology is two-fold: 1) it shows how a network's topology can be exploited to assign an associated dynamic's spectrum along time scales, and 2) it characterizes the performance of the design explicitly in terms of the network's graph topology.

Let us conclude our discussion of the high-gain design, by remarking on several generalizations and connections:

1) The canonical problem studied here provides a simple illustration of the time-scale design strategy, but the strategy applies to other problems. We especially stress that asymmetric partial graph designs can also be addressed.

2) High-gain design can break existing slow-coherency structure (time-scale separation) only when edges between two subnetworks that were initially weakly linked can be designed (and hence the nodes become part of the same partition in our terminology). Our methodology clarifies that such a design not only eliminates slow eigenvalues associated with the coherency structure, but also loses the disturbance-localization properties that result from coherency.

3) Analogous results can be obtained for low-gain designs.

E. An Illustrative Example

We study a 30-node graph (Figure 1). The fixed-edge graph in this graph (identified by the thin blue lines) has edge weights inversely proportional to the length of the line in the plot. The fixed-edge graph has three completely decoupled subgraphs (A,B,and C), and subgraph A itself is composed of two weakly-coupled subgraphs (A_1, A_2). The designable graph (marked by the thick red lines) combines A and B into a single graph, and reduces weak coupling between the subgraphs A_1 and A_2. We are concerned with assigning the spectrum of the graph's Laplacian.

Now let us apply the time-scale design. From Theorem 1, the equivalent plant for this example has a zero dynamics of dimension $R^{25\times 25}$. 23 state variables in the zero dynamics are identical to the state variables of the open-loop plant, while the other two state variables are the averages of the state variables in its partition, e.g., one state variable is the average of the states of vertices 8, 19, 22 and 26. The state matrix of the zero dynamics can be easily characterized in terms of the graph topologies using Theorem 3.

The above structural decomposition provides us with insights into the spectrum of the Laplacian matrix upon high-gain design. The spectrum (or, equivalently, associated dynamics) obtained through modifying these edge weights is constrained by the inherent structure of the zero graph. In this example, as we scale up the weights in the designable edge graph from nominal values, 5 eigenvalues of $L(\Gamma)$ move towards ∞. The other 25 increase monotonically, and approach but can never surpass the zeros, which are in fact the eigenvalues of the scaled zero-graph's Laplacian. From the zero graph, we infer that: 1) one eigenvalue moves from zero to a non-zero value, 2) the original slow-coherent behavior is eliminated/reduced since the zero graph has no edge-cutset of size 1, and 3) many larger eigenvalues change little since they are specified by strongly-connected subgraphs in the fixed-edge graph that are also present in the zero graph. A plot of the 5 smallest non-zero eigenvalues of the Laplacian as we scale the strengths of the 5 designable edges verifies these observations (Figure 1).

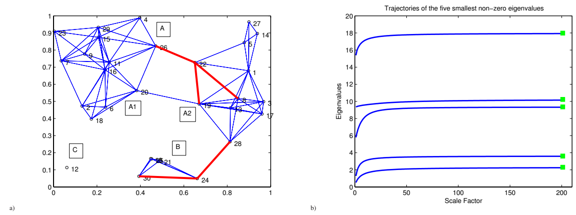

Fig. 1. a) 30-node graph: thin blue lines specify the fixed graph, while the thick red lines specify the designable graph. b) the 5 smallest non-zero eigenvalues of the graph Laplacian as we scale the strengths of the designable communication links.

REFERENCES

[1] A. Saberi and P. Sannuti, "Time-scale structure assignment in linear multivariable systems with high-gain feedback," *International Journal of Control*, vol. 49, no. 6, 1989.

[2] A. Saberi, B. M. Chen, and P. Sannuti, *Loop Transfer Recovery: Analysis and Design*, Springer-Verlag, 1993.

[3] H. K. Ozcetin, A. Saberi, and P. Sannuti, "Almost Disturbance Decoupling Problem with Internal Stability Via State or Measurement Feedback – Singular Perturbation Approach," *International Journal of Control*, vol. 55, no. 4, 1992.

[4] Z. Lin and A. Saberi, "Semi-global exponential stabilization on linear systems subject to "input saturation" via linear feedbacks," *Systems and Control Letters*, vol. 21, 1993.

[5] Y. Wan, S. Roy, A. Saberi, and A. Stoorvogel, "A multiple-derivative and multiple-delay paradigm for decentralized control: introduction using the canonical double-integrator network," in *AIAA 2008 Guidance, Navigation, and Control Conference*, Honolulu, Hawaii, 18-21 August, 2008, also submitted to *Automatica*.

[6] G. N. Ramaswamy, G. C. Verghese, L. Rouco, C. Vialas, and C. DeMarco, "Synchrony, aggregation, and multi-area eigenanalysis," *IEEE Transactions on Power Systems*, vol. 10, no. 4, Nov. 1995.

[7] F. R. K. Chung, *Spectral Graph Theory*, American Mathematical Society Press: Providence, RI, 1997.

[8] S. Boyd, "Convex optimization of graph laplacian eigenvalues," *Proceedings International Congress of Mathematicians*, vol. 3, 2006.

[9] S. Roy and A. Saberi,"Scaling: a Canonical design problem for networks," *International Journal of Control*, vol. 80, no. 8, August 2007.

[10] Y. Wan, S. Roy, and A. Saberi, "A new focus in the science of networks: towards methods for design," *Proceedings of the Royal Society A*, vol. 464, March 2008.

[11] S. Roy, A. Saberi, and K. Herlugson, "Formation and alignment of distributed sensing agents with double-integrator dynamics," IEEE Press Monograph on *Sensor Network Operations*, April 2006.

[12] Y. Wan, S. Roy, and A. Saberi, "Designing spatially-heterogeneous strategies for control of virus spread," *IET Systems Biology*, vol. 2, No. 4, 2008.

[13] P. Sannuti and A. Saberi, "Special coordinate basis for multivariable linear system-finite and infinite zero structure, squaring down and decoupling," *International Journal of Control*, vol. 45, No. 5, 1987.

[14] A. Saberi, P. V. Kokotovic, and H. J. Sussmann, "Global stabilization of partially linear composite systems," *SIAM Journal on Control and Optimization*, vol. 28, no. 6, November 1990.

[15] Y. Wan, S. Roy, X. Wang, A. Saberi, T. Yang, M. Xue, and B. Malek, "On the structure of graph edge designs that optimize the algebraic connectivity," in *Proceedings of 47th IEEE Conference on Decision and Control*, 2008.

[16] S. Roy, Y. Wan, and A. Saberi, "On time-scale designs for networks," to appear in *International Journal of Control*. PDF available at www.eecs.wsu.edu/˜ywan.