



Graph-theoretic optimization for edge consensus

Conference Paper**Author(s):**

Hudoba de Badyn, Mathias ; Foight, Dillon R.; Calderone, Daniel; Mesbahi, Mehran; Smith, Roy 

Publication date:

2021

Permanent link:

<https://doi.org/10.3929/ethz-b-000461441>

Rights / license:

Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

Originally published in:

IFAC-PapersOnLine 54(9), <https://doi.org/10.1016/j.ifacol.2021.06.113>

Graph-Theoretic Optimization for Edge Consensus

Mathias Hudoba de Badyn* Dillon R. Foight**
Daniel Calderone** Mehran Mesbahi** Roy S. Smith*

* Automatic Control Laboratory, ETH Zürich, Switzerland
(e-mails: {mbadyn, rsmith}@control.ee.ethz.ch)

** William E. Boeing Department of Aeronautics & Astronautics,
University of Washington, Seattle WA, 98109 USA
(e-mails: {dfoight, djcal, mesbahi}@uw.edu)

Abstract: We consider network structures that optimize the \mathcal{H}_2 -norm of weighted, time scaled consensus networks, under a minimal representation of such consensus networks described by the edge Laplacian. We show that a greedy algorithm can be used to find the minimum- \mathcal{H}_2 norm spanning tree, as well as how to choose edges to optimize the \mathcal{H}_2 norm when edges are added back to a spanning tree. In the case of edge consensus with a measurement model considering all edges in the graph, we show that adding edges between slow nodes in the graph provides the smallest increase in the \mathcal{H}_2 norm.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Multi-agent systems, consensus, distributed control, \mathcal{H}_2 control

1. INTRODUCTION

Many natural and synthetic systems are distributed among agents in a network. One popular information-sharing protocol distributed over a network is *consensus*, used in many applications ranging from robotics, as in Joordens and Jamshidi (2009); sensor networks, as in Olfati-Saber (2005); and multi-agent systems, as in Olfati-Saber and Murray (2004), and Tanner et al. (2004). Much work on studying networked dynamical systems has focused on how the physical structure of the network affects its dynamics. For example, network symmetries have been shown to be related to controllability of consensus, as discussed by Rahmani et al. (2009) and Alemzadeh et al. (2017).

In this work, we are interested in studying system-theoretic measures of performance, and how to optimize graph structures for these system-theoretic measures. In particular, we examine the \mathcal{H}_2 norm, which can be interpreted as a measure of noise attenuation over the network, as discussed by Siami and Motee (2016), and has a rich history of use in the literature. For consensus networks with leaders, the \mathcal{H}_2 norm has an interpretation in terms of effective resistance across the network, as discussed in Chapman and Mesbahi (2013), and in Hudoba de Badyn and Mesbahi (2019, 2021), a fast method of computing the \mathcal{H}_2 norm for series-parallel networks exploited this interpretation. Bamieh et al. (2012) and Patterson and Bamieh (2010, 2014) have utilized the related concept of coherence to consider local feedback laws and leader selection to promote coherence.

A minimal representation of consensus networks can be obtained by looking at a system where only relative agent states across connections are considered. Such a system,

This work was supported by the Swiss Competence Centers for Energy Research FEEB&D project and the ETH Foundation.

represented by the *edge Laplacian* rather than the *graph Laplacian*, was considered for estimation and control in relative sensing networks in Sandhu et al. (2005, 2009), and for formation flight in Smith and Hadaegh (2005). When considering weighted consensus networks operating with agents that have non-heterogeneous time scales, it was shown in Foight et al. (2019) that edge consensus allows explicit calculations of the \mathcal{H}_2 norm of the system.

This latter formulation is the setting of the present work. In this paper, we focus on the optimization of graph structures in the setting of edge-weighted, and time scaled networks with the \mathcal{H}_2 norm of edge consensus as our optimization measure. The contributions of the paper are as follows. First, we show that a greedy algorithm can be used to find the minimum- \mathcal{H}_2 norm spanning tree. This has applications to systems where resilience to noise is desired, but a minimum number of communication links is also desired, such as in stealthy UAV systems. We then discuss how to add weighted edges back to the graph to improve the \mathcal{H}_2 norm. In the case of edge consensus with a measurement model considering all edges in the graph, we show that adding edges between slow nodes in the graph provides the smallest increase in the \mathcal{H}_2 norm.

The organization of this paper is as follows. In §2, we outline our notation, the mathematical preliminaries on graph and edge Laplacians, and describe our problem statement in the context of the \mathcal{H}_2 performance of edge consensus. Our main results are outlined in §3, and examples are shown in §4. The paper is concluded in §5. Due to space limitations, the proofs are available in the supplementary material Hudoba de Badyn et al. (2020).

2. PRELIMINARIES

In this section, we outline notation and the mathematical preliminaries on graph and systems theory, and then lay out the setting of the problem statement.

2.1 Preliminaries on Graphs & Consensus

We denote the real numbers as \mathbb{R} , the non-negative reals as \mathbb{R}_+ , the positive reals as \mathbb{R}_{++} , and the real n -dimensional Euclidean vector space as \mathbb{R}^n . Vectors in \mathbb{R}^n are written in lower-case x, y, z , etc., and matrices in $\mathbb{R}^{n \times m}$ are written in capital-case M, N, R , etc. I denotes the identity matrix. $\mathbf{1} \in \mathbb{R}^n$ denotes the length- n vector of ones, and $\mathbf{0}$ denotes a matrix of zeros of comfortable dimensions.

A graph \mathcal{G} with n vertices (or nodes) is a triple of sets $(\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = [n]$ is a set of labeled vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of m edges denoting connections between vertices, and $\mathcal{W} \in \mathbb{R}_{++}^m$ is a set of edge weights, denoting some notion of ‘strength’ of the corresponding edge. A graph \mathcal{G} is *undirected* if edge $ij = ji$, and *directed* otherwise. The *neighbourhood* of vertex i is $N_i = \{j : ij \in \mathcal{E}\}$, and the *unweighted degree* of i as $\deg(i) = |N(i)|$.

A *path* \mathcal{P} is a set of edges $\{i_1i_2, i_2i_3, \dots, i_{l-2}i_{l-1}, i_{l-1}i_l\}$ where i_1, \dots, i_l are distinct, and a *cycle* \mathcal{C} is a path, except that the first vertex and the last vertex are the same, i.e. $i_1 = i_l$. An *connected graph* \mathcal{G} is one in which there is an undirected path between any two vertices in \mathcal{G} . A *tree* \mathcal{T} is a connected graph with no cycles, and a *spanning tree* of a connected graph \mathcal{G} is a tree on the same vertex set as \mathcal{G} (i.e., $\mathcal{V}(\mathcal{T}) = \mathcal{V}(\mathcal{G})$), with edges $\mathcal{E}(\mathcal{T}) \subseteq \mathcal{E}(\mathcal{G})$. If \mathcal{T} is a spanning tree of \mathcal{G} , then we can write $\mathcal{T} \subseteq \mathcal{G}$, where ‘ \subseteq ’ is to be interpreted setwise. Note that a tree \mathcal{T} with n vertices must have exactly $n - 1$ edges.

To each vertex $i \in \mathcal{V}$, we assign a state $x_i \in \mathbb{R}$. The dynamics of each vertex state x_i are assumed, unless otherwise stated, to be single-integrator dynamics $\dot{x}_i = u_i$. By setting u_i to be a weighted average of the states of its neighbours, we arrive at the *consensus dynamics*:

$$\dot{x}_i = \sum_{j \in N_i} w_{ij}(x_j - x_i). \quad (1)$$

One can also define the *incidence matrix* $D_{\mathcal{G}} \in \mathbb{R}^{n \times m}$ of an undirected graph \mathcal{G} , where each column d_{ij} of $D_{\mathcal{G}}$ corresponds to an edge $ij \in \mathcal{E}$. The column d_{ij} has 1 in the i th position, and -1 in the j th position; the choice of setting $i = 1$ or $j = 1$ is arbitrary. By setting the matrix $W_{\mathcal{G}} \in \mathbb{R}^{m \times m}$ to be the diagonal matrix containing the edge weights w_{ij} , one can write the (undirected) graph Laplacian as $\mathcal{L}_{\mathcal{G}} = D_{\mathcal{G}}W_{\mathcal{G}}D_{\mathcal{G}}^T$.

By appropriately stacking x_i into a vector $x \in \mathbb{R}^n$, the consensus dynamics in (1) can be written in vector form as as $\dot{x} = -\mathcal{L}_{\mathcal{G}}x$.

A *time-scaled graph* \mathcal{G} is a quadruple of sets $(\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{S})$, where \mathcal{V}, \mathcal{E} , & \mathcal{W} are as before, and $\mathcal{S} \in \mathbb{R}_{++}^n$ is a set of *time scales* $\{\epsilon_i\}_{i=1}^n$ on the vertices. With $E_{\mathcal{G}} = \text{diag}(\{\epsilon_i\}_{i=1}^n)$, we can then define the *time scaled consensus dynamics* as

$$\epsilon_i \dot{x}_i = \sum_{j \in N_i} w_{ij}(x_j - x_i) \iff E_{\mathcal{G}} \dot{x} = -\mathcal{L}_{\mathcal{G}}x.$$

2.2 Edge Consensus and Problem Configuration

The problem considered in this paper is derived from the preceding consensus dynamics, but with process and measurement noise. Consider a graph \mathcal{G} of n multi-scale integrators, and zero-mean Gaussian process noise $\omega_i(t)$,

$$\epsilon_i \dot{x}_i(t) = u_i(t) + \omega_i(t), \quad (2)$$

where $\mathbf{E}[\omega(t)\omega(t)^T] = \text{diag}(\sigma_{\omega_i}^2) \triangleq \Omega^2$, x_i is the (scalar) state of the i -th agent, ϵ_i is node i 's time scale parameter, and u_i is the control input. A weighted, noisy, decentralized feedback controller that seeks to bring agents to consensus is given by,

$$u_i(t) = \sum_{j \in N(i)} [w_{ij}(x_j(t) - x_i(t)) + v_{ij}(t)]$$

$$u(t) = D_{\mathcal{G}}W_{\mathcal{G}}D_{\mathcal{G}}^T x(t) + D_{\mathcal{G}}v(t), \quad (3)$$

where the noise over the edge ij is $v_{ij}(t)$, with covariance $\mathbf{E}[v(t)v(t)^T] = \text{diag}(\sigma_{v_{ij}}^2) \triangleq \Gamma^2$ for all $(i, j) \in \mathcal{E}$. $W_{\mathcal{G}} \triangleq \text{diag}(\mathcal{W})$ is the matrix of edge weights, and $v(t), u(t)$ are the stacked vectors of measurement noises and control inputs. Applying (3) to the stacked-vector version of (2) gives a general, time scaled and weighted consensus dynamics with process and measurement noise,

$$\Sigma_{\mathcal{G}} \triangleq \begin{cases} \dot{x}(t) = E_{\mathcal{G}}^{-1} \mathcal{L}_{\mathcal{G}}x(t) \\ + [E_{\mathcal{G}}^{-1} \quad E_{\mathcal{G}}^{-1} D_{\mathcal{G}}] \begin{bmatrix} \omega(t) \\ v(t) \end{bmatrix}. \end{cases} \quad (4)$$

The scaled Laplacian $E_{\mathcal{G}}^{-1} \mathcal{L}_{\mathcal{G}}$ is rank-deficient (it possesses a zero eigenvalue for each connected component of \mathcal{G}), which complicates reasoning about the \mathcal{H}_2 performance of $\Sigma_{\mathcal{G}}$. Solutions to this problem in the literature include studying the \mathcal{H}_2 norm of the Dirichlet Laplacian, as in Chapman et al. (2015); Hudoba de Badyn and Mesbahi (2019), or as considered in this paper, a minimal representation of $\Sigma_{\mathcal{G}}$ obtained by studying edge consensus, seen in Zelazo et al. (2013); Zelazo and Mesbahi (2011); Foight et al. (2019). In the case of a time scaled graph, the *edge Laplacian* is defined by $\mathcal{L}^{\mathcal{G}} \triangleq D_{\mathcal{G}}^T E_{\mathcal{G}}^{-1} D_{\mathcal{G}}$. We denote the edge Laplacian of a graph \mathcal{G} with a superscript: $\mathcal{L}^{\mathcal{G}}$, and the graph Laplacian with a subscript: $\mathcal{L}_{\mathcal{G}}$.

In Zelazo and Mesbahi (2011), it was shown that the graph and edge Laplacians are related by a similarity transformation; Foight et al. (2019) extended the transformation to time-scaled and matrix-weighted graphs. This similarity transformation is constructed by choosing a spanning tree $\mathcal{T} \subseteq \mathcal{G}$, and noting that the cycles can be considered as linear combinations of the spanning tree edges in the following sense. Let $D_{\mathcal{T}}$ denote the incidence matrix of the spanning tree \mathcal{T} , and let $D_{\mathcal{G} \setminus \mathcal{T}}$ denote the incidence matrix of the remaining edges that correspond to the edges completing the cycles in \mathcal{G} . Define the *Tucker representation matrix* $R_{\mathcal{G}} \triangleq [I \quad T_{\mathcal{T}}]$, where $T_{\mathcal{T}} = [D_{\mathcal{T}}^T D_{\mathcal{T}}]^{-1} D_{\mathcal{T}}^T D_{\mathcal{G} \setminus \mathcal{T}}$. We refer to the matrix $T_{\mathcal{T}}$ as the *cycle representation matrix*. Note that if the graph is a tree, i.e. $\mathcal{G} = \mathcal{T}$, then the only choice of spanning tree for $R_{\mathcal{G}}$ is in fact \mathcal{T} , and $T_{\mathcal{T}}$ is the empty matrix ($R_{\mathcal{G}} = I$ in this case).

It suffices to measure the relative states of the nodes $x_i - x_j$ over the edges ij of a spanning tree $\mathcal{T} \subseteq \mathcal{G}$ to capture consensus: if $x_i - x_j = 0$ for all $ij \in \mathcal{T}$, then we know that the network is at consensus. This is the motivation for the edge consensus model, formalized below. We define the *edge states* for a edge weighted and time scaled graph as $x_e \triangleq S_v^{-1}x$, where the similarity transformation and corresponding state matrix are,

$$S_v = \begin{bmatrix} E_{\mathcal{G}}^{-1} D_{\mathcal{T}} [D_{\mathcal{T}}^T E_{\mathcal{G}}^{-1} D_{\mathcal{T}}]^{-1} & \mathbf{1} \end{bmatrix}$$

$$S_v^{-1} \mathcal{L}_{\mathcal{G}} S_v = \begin{bmatrix} \mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{L}^{\mathcal{T}} = D_{\mathcal{T}}^{\mathcal{T}} E_{\mathcal{G}}^{-1} D_{\mathcal{T}}.$$

The dynamics of the edge states x_e corresponding to (4) are then given by

$$\dot{x}_e = \begin{bmatrix} -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} & 0 \\ 0 & 0 \end{bmatrix} x_e + \begin{bmatrix} D_{\mathcal{T}}^{\mathcal{T}} E_{\mathcal{G}}^{-1} & -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} \\ \text{tr}[E_{\mathcal{G}}]^{-1} \mathbf{1}^{\mathcal{T}} & 0 \end{bmatrix} \begin{bmatrix} \omega(t) \\ v(t) \end{bmatrix}. \quad (5)$$

The form of the system matrix in (5) suggests a partitioning of x_e into the edge states of the chosen spanning tree \mathcal{T} and, and the consensus subspace, i.e. $x_e = [x_{\mathcal{T}}^{\mathcal{T}} \quad x_{\mathbf{1}}]^{\mathcal{T}}$. Finally, we arrive at two models of the spanning tree edge states, differing in their choice of output:

$$\Sigma^{\mathcal{T}}[\mathcal{G}] \triangleq \begin{cases} \dot{x}_{\mathcal{T}} & = -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} x_{\mathcal{T}} + D_{\mathcal{T}}^{\mathcal{T}} E_{\mathcal{G}}^{-1} \omega \\ & -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} v \\ z & = R_{\mathcal{G}}^{\mathcal{T}} x_{\mathcal{T}} \end{cases} \quad (6)$$

$$\hat{\Sigma}^{\mathcal{T}}[\mathcal{G}] \triangleq \begin{cases} \dot{x}_{\mathcal{T}} & = -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} x_{\mathcal{T}} + D_{\mathcal{T}}^{\mathcal{T}} E_{\mathcal{G}}^{-1} \omega \\ & -\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} v \\ z & = x_{\mathcal{T}}. \end{cases} \quad (7)$$

There are multiple motivations for each of the above models, and the edge consensus formulation in general. As we can see in (5), the edge state dynamics naturally exclude the zero eigenvalue of $\mathcal{L}_{\mathcal{G}}$. Also, as proposed in Zelazo and Mesbahi (2011), it suffices to measure the $n-1$ relative states across the edges of a spanning tree \mathcal{T} to determine if the states of the n nodes are at consensus. Therefore, a satisfactory output measurement of the network is precisely the edge states of that spanning tree as in (7). Of course, the remaining edges that are not in \mathcal{T} also influence the dynamics, and to study the \mathcal{H}_2 norm of the entire system one can take the output of these edge states into account as well. The model in (6) has output $z = R_{\mathcal{G}}^{\mathcal{T}} x_{\mathcal{T}}$, which reconstructs the states of the remaining edges in $\mathcal{G} \setminus \mathcal{T}$ using the Tucker representation matrix, and thus considers all edges in the output.

In the next section, we discuss the \mathcal{H}_2 norm as a system-theoretic metric of performance of these models. For brevity, we may drop the argument of $\Sigma^{\mathcal{T}}[\mathcal{G}]$.

2.3 Systems Theory and \mathcal{H}_2 Performance

The input-output excitation properties of a linear system $\dot{x} = Ax + Bu$, $y = Cx$ with transfer function $G(s) = C(sI - A)^{-1}B$ can be described using the \mathcal{H}_2 norm,

$$\mathcal{H}_2^2(G) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{tr}[G(j\omega)^* G(j\omega)] d\omega.$$

The \mathcal{H}_2 norm measures the steady-state covariance of the output of the system under zero-mean, unit-covariance white noise inputs, or equivalently the root-mean-square of the impulse response of the system.

As discussed in Foight et al. (2019), the \mathcal{H}_2 norms of the models $\Sigma^{\mathcal{T}}[\mathcal{G}]$, $\hat{\Sigma}^{\mathcal{T}}[\mathcal{G}]$ in (6) and (7) are obtained by solving the Lyapunov equation

$$-\mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} X - X R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}} \mathcal{L}^{\mathcal{T}} + D_{\mathcal{T}}^{\mathcal{T}} E_{\mathcal{G}}^{-1} \Omega^{\mathcal{T}} E_{\mathcal{G}}^{-1} D_{\mathcal{T}} + \mathcal{L}^{\mathcal{T}} R_{\mathcal{G}} \Gamma \Gamma^{\mathcal{T}} R_{\mathcal{G}}^{\mathcal{T}} \mathcal{L}^{\mathcal{T}} = 0. \quad (8)$$

An analytic solution to Equation (8) can be obtained with a convenient choice of the error covariances Ω and Γ ,

specifically $\Omega = \sigma_{\omega} E_{\mathcal{G}}^{1/2}$ and $\Gamma = \omega_v W_{\mathcal{G}}^{1/2}$. In this case, the solution of (8) is $X^* = \frac{1}{2} [\sigma_{\omega}^2 (R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}})^{-1} + \sigma_v^2 \mathcal{L}^{\mathcal{T}}]$, and so the \mathcal{H}_2 norms of $\Sigma^{\mathcal{T}}[\mathcal{G}]$, $\hat{\Sigma}^{\mathcal{T}}[\mathcal{G}]$ are, respectively,

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^{\mathcal{T}}[\mathcal{G}]) &= \text{tr}[R_{\mathcal{G}}^{\mathcal{T}} X^* R_{\mathcal{G}}] \\ &= \frac{\sigma_{\omega}^2}{2} \text{tr}[R_{\mathcal{G}}^{\mathcal{T}} (R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}})^{-1} R_{\mathcal{G}}] + \frac{\sigma_v^2}{2} \text{tr}[R_{\mathcal{G}}^{\mathcal{T}} \mathcal{L}^{\mathcal{T}} R_{\mathcal{G}}] \quad (9) \\ \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}[\mathcal{G}]) &= \text{tr}[X^*] \\ &= \frac{\sigma_{\omega}^2}{2} \text{tr}[(R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}})^{-1}] + \frac{\sigma_v^2}{2} \text{tr}[\mathcal{L}^{\mathcal{T}}]. \quad (10) \end{aligned}$$

Of course, the choice of the error covariances causes a loss of generality in the applicability of the analysis of this model. In Foight et al. (2019, 2020), it was shown that this choice, plus the true error covariances can be combined with an ordering in the semi-definite cone of covariances to reasonably bound the true performance, suggesting that this model is a convenient proxy for general noise models.

It was noted in Foight et al. (2019) that the contributions of the weights and time scales can be separated in (9) & (10) as,

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^{\mathcal{T}}[\mathcal{G}]) &= \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; W) + \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; E) \\ \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}[\mathcal{G}]) &= \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; W) + \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; E) \end{aligned}$$

where

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; W) &\triangleq \frac{\sigma_{\omega}^2}{2} \text{tr}[R_{\mathcal{G}}^{\mathcal{T}} (R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}})^{-1} R_{\mathcal{G}}] \\ \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; W) &\triangleq \frac{\sigma_{\omega}^2}{2} \text{tr}[(R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}}^{\mathcal{T}})^{-1}] \\ \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; E) &\triangleq \frac{\sigma_v^2}{2} \text{tr}[R_{\mathcal{G}}^{\mathcal{T}} \mathcal{L}^{\mathcal{T}} R_{\mathcal{G}}] \\ \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; E) &\triangleq \frac{\sigma_v^2}{2} \text{tr}[\mathcal{L}^{\mathcal{T}}]. \end{aligned}$$

Furthermore, it was shown in Foight et al. (2019) that the \mathcal{H}_2 norm of Σ can be written in terms of the \mathcal{H}_2 norm of $\hat{\Sigma}$ as follows:

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; W) &= \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; W) + \frac{\sigma_{\omega}^2}{2} \text{tr}[T_{\mathcal{T}}^{\mathcal{T}} (R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}})^{-1} T_{\mathcal{T}}] \quad (11) \end{aligned}$$

$$\mathcal{H}_2^2(\Sigma^{\mathcal{T}}; E) = \mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}}; E) + \frac{\sigma_v^2}{2} \text{tr}[T_{\mathcal{T}}^{\mathcal{T}} L^{\mathcal{T}} T_{\mathcal{T}}].$$

We examine the behaviour of each of these components when an edge is added to a spanning tree – $\mathcal{H}_2^2(\hat{\Sigma}, W)$ and $\mathcal{H}_2^2(\hat{\Sigma}, E)$ are discussed in §3.2, and $\mathcal{H}_2^2(\Sigma, E)$ & $\mathcal{H}_2^2(\Sigma, W)$ in §3.3 and §3.4 respectively.

3. MAIN RESULTS

3.1 Minimum- \mathcal{H}_2 Spanning Tree

In this section, we show how a minimum- \mathcal{H}_2 norm spanning tree of a graph \mathcal{G} can be found with a greedy algorithm. Let \mathcal{G} be a weighted, time scaled graph with edge consensus dynamics (7) and corresponding \mathcal{H}_2 norm (10).

Suppose that we want to remove communication links from \mathcal{G} until we are left with a minimally-connected graph — a tree. Such an operation can be used during a ‘stealth mode’ for networked systems, during which time having the minimum number of communication links while still being connected is desired. For example, a swarm of UAVs

may be using consensus to agree on a formation heading, but may want to limit their chance of detection when entering a hostile area. In this section, we show that one can choose the spanning tree $\mathcal{T} \subseteq \mathcal{G}$ that minimizes the \mathcal{H}_2 norm in (10) using a greedy algorithm. This allows the network to minimize the number of communication links used in the consensus algorithm, while maintaining optimal noise rejection properties. The main result of this section is summarized in Proposition 1.

Algorithm 1 Minimum- \mathcal{H}_2 Spanning Tree

```

1: Input:  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{S})$ 
2: Initialize:
3: Choose  $v \in \mathcal{V}$ , set  $\mathcal{Q} = \mathcal{V} \setminus \{v\}$ ,  $\mathcal{P} = \{v\}$ ,  $\mathcal{R} = \emptyset$ 
4: while  $|\mathcal{Q}| > 0$  do
5:   Set
       $N(\mathcal{P}) = \{j \in \mathcal{Q} : i \in \mathcal{P}, ij \in \mathcal{E}\}$ .
6:   Solve
      (P1) :=  $\min_{j \in N(\mathcal{P})} \left\{ \sigma_v^2(\epsilon_k^{-1} + \epsilon_j^{-1}) + \sigma_\omega^2 w_{jk}^{-1} : jk \in \mathcal{E} \right\}$ 
7:   From  $\arg \min$  (P1) :=  $(j^*, j^*k)$ , update
       $\mathcal{P} \mapsto \mathcal{P} \cup \{j^*\}$ 
       $\mathcal{R} \mapsto \mathcal{R} \cup \{j^*k\}$ 
       $\mathcal{Q} \mapsto \mathcal{Q} \setminus \{j^*\}$ 
8: end while
9: Output:  $\mathcal{T}^* = (\mathcal{P}, \mathcal{R}, \mathcal{W}(\mathcal{R}), \mathcal{S})$ 

```

Proposition 1. Consider a graph \mathcal{G} with the edge consensus dynamics (7), and with corresponding \mathcal{H}_2 norm (10). Any spanning tree $\mathcal{T}' \subseteq \mathcal{G}$ has \mathcal{H}_2 norm

$$\mathcal{H}_2^2(\hat{\Sigma}^{\mathcal{T}'}) = \frac{\sigma_\omega^2}{2} \sum_{e \in \mathcal{T}'} w_e^{-1} + \frac{\sigma_v^2}{2} \sum_{i=1}^n \frac{\deg(i)}{\epsilon_i}. \quad (12)$$

Algorithm 1 returns a spanning tree $\mathcal{T}^* \subseteq \mathcal{G}$ that minimizes (12) out of all possible spanning trees $\mathcal{T} \subseteq \mathcal{G}$.

Remark 2. For an edge-weighted graph where the cost is the sum of the edge weights, it is known that if the edge weights are distinct, then the minimal-weighted spanning tree is unique. The minimum- \mathcal{H}_2 spanning tree \mathcal{T}^* generated by Algorithm 1 is not unique, even if all the nodal time scales and edge weights are distinct. See §4.1 for an example.

Remark 3. Another practical algorithm would be a distributed version of Algorithm 1, which could be constructed by considering a distributed minimal-weighted spanning tree algorithm such as Awerbuch (1987); Gallager et al. (1983); Kutten and Peleg (1998). Such an algorithm would allow a consensus network to autonomously restructure itself to the \mathcal{H}_2 -optimal tree configuration in a distributed manner.

3.2 Weighted Cycle Selection

In the previous section, we discussed how one may find the minimum- \mathcal{H}_2 norm spanning tree from a connected graph \mathcal{G} . In this section, we examine what happens to the \mathcal{H}_2 norm when edges are added to a spanning tree. In particular, we show that adding weighted edges improve the \mathcal{H}_2 norm, and discuss what choice of edge optimizes this improvement. This can be used in the previous motivating example of a stealthy system that wants to still minimize the number of communication links, but does not reject noise well enough with just the links in the spanning tree.

Zelazo et al. (2013) discussed how adding cycles to an unweighted, mono-scaled graph impacts the \mathcal{H}_2 performance in the presence of unit covariance noise applied to the edges and nodes. In particular, they found that the biggest improvement in \mathcal{H}_2 performance resulted from adding an edge to a tree \mathcal{T} which maximized the length of the resulting cycle. Our first contribution is showing that in the presence of edge weights and time scales, the length of the cycle no longer matters, rather one should add an edge to form a cycle that, roughly speaking, has small weights.

First, we define the *weighted length* and *unweighted length* of a cycle \mathcal{C} as, $l_w(\mathcal{C}) := \sum_{e \in \mathcal{C}} w_e^{-1}$, $l(\mathcal{C}) := \sum_{e \in \mathcal{C}} 1$.

Proposition 4. Consider a graph consisting of a weighted, time scaled tree \mathcal{T} , and consider the dynamics $\hat{\Sigma}^{\mathcal{T}}$, with the \mathcal{H}_2 norm of $\hat{\Sigma}^{\mathcal{T}}$ given by Equation (10). Consider the task of adding edges to \mathcal{T} to optimize the \mathcal{H}_2 norm of $\hat{\Sigma}^{\mathcal{T}}$. Then, the following hold:

(1) Adding an edge e with weight W_C to \mathcal{T} improves the \mathcal{H}_2 performance in the following manner:

$$\mathcal{H}_2^2(\hat{\Sigma}[\mathcal{T} \cup e]) = \mathcal{H}_2^2(\hat{\Sigma}[\mathcal{T}]) - \frac{\sigma_\omega^2}{2l_w(\mathcal{C})} \sum_{ij \in \mathcal{T} \cap \mathcal{C}} w_{ij}^{-2},$$

where \mathcal{C} is the unique cycle formed by adding edge e to \mathcal{T} . Furthermore, adding an edge to \mathcal{T} to form a cycle always decreases the \mathcal{H}_2 norm.

(2) Adding p edge-disjoint cycles $\mathcal{C}_1, \dots, \mathcal{C}_p$ via edges e_1, \dots, e_p improves the \mathcal{H}_2 performance in the following manner:

$$\begin{aligned} \mathcal{H}_2^2(\hat{\Sigma}[\mathcal{T} \cup \{e_i\}]) \\ = \mathcal{H}_2^2(\hat{\Sigma}[\mathcal{T}]) - \sum_{k=1}^p \left[\frac{\sigma_\omega^2}{2l_w(\mathcal{C}_k)} \sum_{ij \in \mathcal{T} \cap \mathcal{C}_k} w_{ij}^{-2} \right], \end{aligned}$$

3.3 Contributions of Time Scales to $\mathcal{H}_2^2(\Sigma^{\mathcal{T}}[\mathcal{T}])$

As discussed in §2.3, the two models $\Sigma^{\mathcal{T}}$, $\hat{\Sigma}^{\mathcal{T}}$ differ in the outputs – the former considers only the edge states of the spanning tree, and the latter considers all edges of the graph. We noted in the proof of Proposition 1 that when adding an edge to a tree, the \mathcal{H}_2 norm of the dynamics $\hat{\Sigma}^{\mathcal{T}}$ is not affected by the time scales of the nodes on which the edge is placed. In this section, we show that these dynamics, adding an edge to slow nodes provides the smallest increase in \mathcal{H}_2 norm.

Recall that the \mathcal{H}_2 performance of $\Sigma^{\mathcal{T}}$ can be written as,

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^{\mathcal{T}}) &= \frac{\sigma_\omega^2}{2} \mathbf{tr}[R_G^T (R_G W_G R_G^T)^{-1} R] + \frac{\sigma_v^2}{2} \mathbf{tr}[R_G^T \mathcal{L}^T R_G] \\ &:= \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; W) + \mathcal{H}_2^2(\Sigma^{\mathcal{T}}; E). \end{aligned} \quad (13)$$

Here we have separated the contributions of the weights and time scales into two terms. Consider the second term in (13), defined as

$$\mathcal{H}_2^2(\Sigma^{\mathcal{T}}; E) := \frac{1}{2} \sigma_v^2 \mathbf{tr}[R_G^T \mathcal{L}^T R_G].$$

We have the following proposition.

Proposition 5. There exists a separation of $\mathcal{H}_2(\Sigma, E)$ into two terms, one depending on the cycle states, and the other the spanning tree states. Namely,

$$\mathcal{H}_2^2(\Sigma; E) = \frac{\sigma_v^2}{2} \left(\mathbf{tr}[\mathcal{L}^{\mathcal{T}}] + \mathbf{tr}[\mathcal{L}^{\mathcal{G} \setminus \mathcal{T}}] \right).$$

From Proposition 5, we can conclude with the following corollary.

Corollary 6. Consider the dynamics (6) of Σ^T on a graph \mathcal{G} . Then, consider adding an edge ij to \mathcal{G} to get $\mathcal{G} \cup \{ij\}$. The time scale term of the \mathcal{H}_2^2 norm in (13) satisfies

$$\mathcal{H}_2^2(\Sigma^T[\mathcal{G} \cup \{ij\}]; E) = \frac{\sigma_v^2}{2} (\epsilon_i^{-1} + \epsilon_j^{-1}) + \mathcal{H}_2^2(\Sigma^T[\mathcal{G}], E).$$

3.4 Contribution of Weights to $\mathcal{H}_2^2(\Sigma^T[\mathcal{T}])$

In §3.2 we showed that adding an edge to a spanning tree always improves the \mathcal{H}_2 norm of $\hat{\Sigma}$. In this section, we discuss what happens to \mathcal{H}_2 norm of Σ by examining (11).

Recall from §2.3 that we can write

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^T; W) &= \mathcal{H}_2^2(\hat{\Sigma}^T; W) \\ &+ \frac{\sigma_w^2}{2} \text{tr} [T_{\mathcal{T}}^T (R_{\mathcal{G}} W_{\mathcal{G}} R_{\mathcal{G}})^{-1} T_{\mathcal{T}}]. \end{aligned} \quad (14)$$

When $\mathcal{G} = \mathcal{T}$, we examined what happens to $\mathcal{H}_2^2(\hat{\Sigma}^T; W)$ when an edge is added to \mathcal{T} in Proposition 4. When $\mathcal{G} = \mathcal{T}$, the second term in (14) is zero since $T_{\mathcal{T}}$ is the empty matrix. In the next proposition, we show that this term is positive when an edge is added to \mathcal{T} .

Proposition 7. Consider the edge consensus model (6) on a graph \mathcal{T} with \mathcal{H}_2 norm (9). When an edge e with weight W_C is added to \mathcal{T} forming a cycle \mathcal{C} , the weight portion of the \mathcal{H}_2 norm satisfies

$$\begin{aligned} \mathcal{H}_2^2(\Sigma^T[\mathcal{T} \cup \{ij\}]; W) &= \mathcal{H}_2^2(\Sigma^T[\mathcal{T}]; W) \\ &+ \frac{\sigma_w^2}{2W_C} - \frac{\sigma_w^2}{2l_w(\mathcal{C})} \sum_{ij \in \mathcal{C}} w_{ij}^{-2}. \end{aligned}$$

4. EXAMPLES

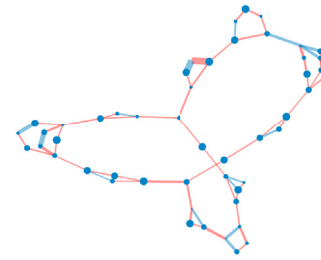
4.1 Minimum \mathcal{H}_2 -Norm Spanning Trees

In this section, we show examples of the minimum- \mathcal{H}_2 spanning trees from Algorithm 1. Three such graphs are shown in Figure 1; edges in the minimum- \mathcal{H}_2 spanning tree are highlighted in red, and the remaining rejected edges are highlighted in blue. In each example, the edge weights w_e and time scales ϵ_i are drawn from a uniform random distribution on $[0, 1]$ — the width of each edge is proportional to $\log(w_e)$ for clarity, and the node size is proportional to ϵ_i . The graph in Figure 1a is an example of a series-parallel network; the graph in Figure 1b is the molecular adjacency structure of the caffeine molecule, and the graph in Figure 1c is an Erdős-Rényi random graph. The random graph is generated on 40 nodes, and an edge between node i and j exists with probability 0.09.

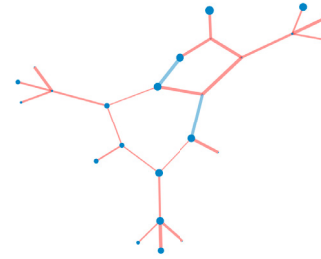
Next, we show an example of a graph with distinct time scales on the nodes, and distinct edge weights, but no unique minimum- \mathcal{H}_2 spanning tree. Consider the triangle in Figure 2 with the edge consensus model $\hat{\Sigma}^T$, and the following time scales and edge weights:

$$\epsilon_1 = 1, \epsilon_2 = 2, \epsilon_3 = 3 \quad w_{23} = 1, \quad w_{13} = 2, \quad w_{12} = 3.$$

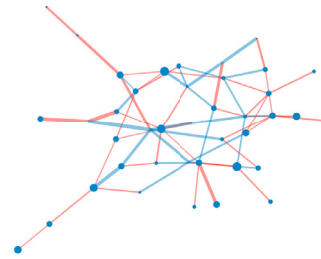
Further suppose that the noise weightings satisfy $\sigma_w = \sigma_v = 1$. Then, the spanning trees \mathcal{T}_1 and \mathcal{T}_2 of the triangle in Figure 2 on the right have \mathcal{H}_2 norms $\mathcal{H}_2(\Sigma^{\mathcal{T}_1}(\mathcal{G})) = \mathcal{H}_2(\Sigma^{\mathcal{T}_2}(\mathcal{G})) = 1.8\bar{3}$.



(a) Series-parallel network



(b) Adjacency graph of the caffeine molecule



(c) Erdős-Rényi random graph with $P(ij \in \mathcal{E}) = 0.09$

Fig. 1. Example graphs with the minimum- \mathcal{H}_2 norm spanning tree (highlighted in red), computed from Algorithm 1. Rejected edges from the original graph are highlighted in blue. Edge widths are proportional to $\log(w_e)$, and node markers are proportional to ϵ_i .

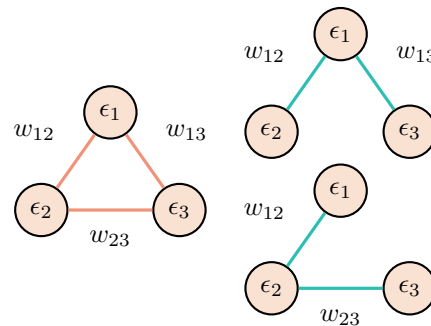


Fig. 2. Example of a time scaled, weighted graph with distinct edge weights and time scales, but no unique minimal \mathcal{H}_2 spanning tree.

4.2 Adding Edges to a Tree

Consider the path graph in Figure 3 denoted by the solid edges, where all the edges in the path have weight $w_e = 1$. Suppose $W_1 = 10$ and $W_2 = 5$, and $\sigma_w = 1$. Then, the improvement of adding either edge W_1 to \mathcal{G} to create cycle \mathcal{C}_1 , or adding W_2 to \mathcal{G} to create cycle \mathcal{C}_2 is

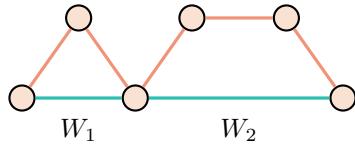


Fig. 3. Example of a weighted graph where a cycle with fewer edges has a lower \mathcal{H}_2 norm for $\hat{\Sigma}^T$ than a cycle with more edges.

$$-\frac{1}{2l_w(C_i)} \left[\sum_{e \in T \cap C_i} w_e^{-2} \right] \approx \begin{cases} -0.9524 & \text{if } i = 1 \\ -0.9375 & \text{if } i = 2, \end{cases}$$

and so the shorter (but more highly weighted) cycle C_1 offers the better improvement.

5. CONCLUSION

In this paper, we have considered various perspectives on optimizing weights and time scales for edge consensus. We showed that a greedy algorithm can be used to find the minimum- \mathcal{H}_2 norm spanning tree of a graph, and then discussed how the \mathcal{H}_2 norm can be optimized by adding cycles for two output models of edge consensus.

REFERENCES

- Alemzadeh, S., Hudoba de Badyn, M., and Mesbahi, M. (2017). Controllability and stabilizability analysis of signed consensus networks. In *Proc. IEEE Conference on Control Technology and Applications*, 55–60. Kohala Coast, USA.
- Awerbuch, B. (1987). Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election and Related Problems. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 230–240.
- Bamieh, B., Ivanović, M.R., Mitra, P., and Patterson, S. (2012). Coherence in Large-Scale Networks: Dimension-Dependent Limitations of Local Feedback. *IEEE Transactions on Automatic Control*, 57(9), 2235–2249.
- Chapman, A. and Mesbahi, M. (2013). Semi-autonomous consensus: Network measures and adaptive trees. *IEEE Transactions on Automatic Control*, 58(1), 19–31.
- Chapman, A., Schoof, E., and Mesbahi, M. (2015). Online Adaptive Network Design for Disturbance Rejection. In *Principles of Cyber-Physical Systems*. Cambridge University Press.
- Foight, D.R., Hudoba de Badyn, M., and Mesbahi, M. (2019). Time Scale Design for Network Resilience. In *Proc. 58th IEEE Conference on Decision and Control*, 2096–2101. Nice, France.
- Foight, D.R., Hudoba de Badyn, M., and Mesbahi, M. (2020). Performance and design of consensus on matrix-weighted and time scaled graphs. *IEEE Transactions on Control of Network Systems*, 7(4), 1812–1822.
- Gallager, R.G., Humblet, P.A., and Spira, P.M. (1983). A Distributed Algorithm for Minimum-Weight Spanning Trees. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 5(1), 66–77.
- Hudoba de Badyn, M., Foight, D.R., Calderone, D., Mesbahi, M., and Smith, R.S. (2020). Graph-theoretic optimization for edge consensus. In *24th International Symposium on the Mathematical Theory of Networks and Systems*. Cambridge, United Kingdom. URL <http://arxiv.org/abs/2006.16201>.
- Hudoba de Badyn, M. and Mesbahi, M. (2019). Efficient Computation of Performance on Series-Parallel Networks. In *Proc. American Control Conference*, 3364–3369. Philadelphia, USA.
- Hudoba de Badyn, M. and Mesbahi, M. (2021). H2 Performance of Series-Parallel Networks: A Compositional Perspective. *IEEE Transactions on Automatic Control*, 66(1), 354–361.
- Joordens, M.A. and Jamshidi, M. (2009). Underwater swarm robotics consensus control. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, October, 3163–3168. San Antonio, USA.
- Kutten, S. and Peleg, D. (1998). Fast distributed construction of small k-dominating sets and applications. *Journal of Algorithms*, 28(1), 40–66.
- Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, 8179–8184. Seville, Spain.
- Olfati-Saber, R. and Murray, R.M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.
- Patterson, S. and Bamieh, B. (2010). Leader selection for optimal network coherence. *Proceedings of the IEEE Conference on Decision and Control*, 2692–2697.
- Patterson, S. and Bamieh, B. (2014). Consensus and coherence in fractal networks. *IEEE Transactions on Control of Network Systems*, 1(4), 338–348.
- Rahmani, A., Ji, M., Mesbahi, M., and Egerstedt, M. (2009). Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1), 162–186.
- Sandhu, J., Mesbahi, M., and Tsukamaki, T. (2005). Relative sensing networks: Observability, estimation, and the control structure. In *Proc. 44th IEEE Conference on Decision and Control, and the European Control Conference*, 6400–6405. Seville, Spain.
- Sandhu, J., Mesbahi, M., and Tsukamaki, T. (2009). On the control and estimation over relative sensing networks. *IEEE Transactions on Automatic Control*, 54(12), 2859–2863.
- Siemi, M. and Motee, N. (2016). Fundamental limits and tradeoffs on disturbance propagation in linear dynamical networks. *IEEE Transaction on Automatic Control*, 61(12), 4055–4062.
- Smith, R.S. and Hadaegh, F.Y. (2005). Control of deep-space formation-flying spacecraft; relative sensing and switched information. *Journal of Guidance, Control, and Dynamics*, 28(1), 106–114.
- Tanner, H.G., Pappas, G.J., and Kumar, V. (2004). Leader-to-formation stability. *IEEE Transactions on Robotics and Automation*, 20(3), 443–455.
- Zelazo, D. and Mesbahi, M. (2011). Edge agreement: Graph-theoretic performance bounds and passivity analysis. *IEEE Transactions on Automatic Control*, 56(3), 544–555.
- Zelazo, D., Schuler, S., and Allgöwer, F. (2013). Performance and design of cycles in consensus networks. *Systems and Control Letters*, 62(1), 85–96.