Performance Improvement in Noisy Linear Consensus Networks With Time-Delay

Yaser Ghaedsharaf, *Student Member, IEEE*, Milad Siami, *Member, IEEE*, Christoforos Somarakis, *Member, IEEE*, and Nader Motee, *Senior Member, IEEE*

***Abstract*—We analyze performance of a class of time-delay first-order consensus networks from a graph topological perspective and present methods to improve it. Performance is measured by network's square of $\mathcal{H}_2$ norm and it is derived in closed form. Moreover, we prove that performance is a convex function of the coupling weights of the underlying graph. We demonstrate that the effect of time-delay reincarnates itself in the form of non-monotonicity, leading to counter-intuitive behaviors of the performance as a function of graph topology. For the network design problem, we propose a tight but simple approximation of the performance measure in order to achieve lower complexity in our problems by eliminating the computationally expensive need for eigendecomposition. More specifically, we discuss three $\mathcal{H}_2$-based optimal design methods to enhance performance. The proposed algorithms provide near-optimal solutions with improved computational complexity as opposed to existing methods in the literature.**

***Index Terms*—Approximation methods, time-delay systems, greedy algorithms, multi-agent systems, network analysis and control, network growing, sparsification.**

I. Introduction

Our objective is to characterize $\mathcal{H}_2$-norm performance of a noisy time-delay linear consensus network using the spectrum of the Laplacian matrix, quantify inherent fundamental limits on its best achievable performance, and eventually develop low time-complexity and efficient algorithms to improve consensus network performance.

Measures for performance of consensus networks and the problem of designing such networks to achieve optimal performance have been extensively studied in the past decades. Performances of consensus networks in the absence of time-delay were studied in [1]–[4]. Optimal design of averaging networks was studied in [5]. Minimizing the total effective resistance of the graph was investigated in [6]. By using the fact that the total effective resistance of an underlying graph is proportional to the $\mathcal{H}_2$-norm square of first-order consensus networks, the works in [7], [8] consider to improve the performance measure by growing the underlying graph. Siami and Motee [9] establish the relation between algebraic connectivity of the coupling graph of the network and performance of linear consensus networks in the absence of time-delay. NP-hardness of the problem of adding a prespecified number of edges to a network via maximizing its algebraic connectivity is proven in [10], whereas Ghosh *et al.* [6] suggested a heuristic using the Fiedler vector of the graph to address the problem. Despite extensive study of networks' performance in the absence of time-delay, limited attention has been given to the performance analysis of linear consensus networks in the presence of time-delay.

The stability analysis of first-order linear consensus networks with homogeneous time-delay and undirected couplings was studied in [9]. Necessary and sufficient conditions for stability of a linear undirected network with nonuniform delay were reported in [11]. Scalable condition for robust stability of networks with heterogeneous stable dynamics using *S*-hulls was considered in [12], [13]. Jönsson [13] propose a low-complexity stability criterion for a class of large-scale systems and a scalable robust stability criterion for interconnected systems with heterogeneous linear time-invariant subsystems is reported in [14]. The stability analysis of consensus and oscillatory nonlinear networks as well as switching networks with heterogeneous time-delays is tackled in [15]. Moreover, a unifying framework to analyze network synchronization using integral quadratic constraints is proposed in [16]. In [17], Somarakis and Baras study convergence rate of averaging networks subject to multiple time-dependent delays. Qiao and Sipahi [18] aim at designing a robustly stable network with respect to time-delay, and in [19], Rafiee and Bayen maximize algebraic connectivity of an underlying graph of a time-delay linear consensus network; nonetheless, importance and relevance of algebraic connectivity in performance evaluation of time-delay networks remain disputable. Finally, the works in [20]–[22] analyze synchronization efficiency in a first-order consensus network with uniform or multiple time-delay. Although most of are limited to numerical results, some of them coincide with our work in derivation of the performance measure in a special case, i.e., when the output matrix is a centering matrix.

The purpose of this manuscript is to analyze performance of time-delay linear consensus networks and propose design algorithms to achieve the best attainable performance for such networks. We assume that all time-delays are identical and coupling (communication) graphs of networks are

Manuscript received June 19, 2017; revised February 13, 2018 and February 15, 2018; accepted July 18, 2018. Date of publication October 8, 2018; date of current version May 27, 2019. This work was supported in part by the NSF CAREER ECCS-1454022 and in part by the ONR YIP N00014-16-1-2645. Recommended by Associate Editor C.-Y. Kao. *(Corresponding author: Nader Motee.)*

Y. Ghaedsharaf, C. Somarakis, and N. Motee are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015 USA (e-mail: ghaedsharaf@lehigh.edu; csomarak@lehigh.edu; motee@lehigh.edu).

M. Siami is with the Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: siami@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2018.2874675

undirected (bidirectional). These assumptions allow us to quantify performance explicitly by deriving a closed form expression. Relaxing these assumptions to include networks with nonuniform time-delays and/or directed coupling graphs is challenging and requires separate investigations, which may not lend themselves to analytic performance evaluation and prevent us from devising low time-complexity algorithms [23], [24]. Since time-delay is intrinsic to all networked control systems, devising efficient and scalable algorithms for analysis and design of networked systems with higher order dynamics and time-delay will remain an active research area, for many years to come.

This work is an outgrowth of [25]–[27], and presents a consistent story on how to analyze and improve the performance of noisy linear consensus networks subject to time-delay. In [25], we studied properties of the first-order consensus network's $\mathcal{H}_2$-norm in the presence of time-delay, and in [26] and [27], we offered growing and sparsification algorithms to enhance the $\mathcal{H}_2$-norm of the network. This manuscript extends results of [25]–[27] to the general output matrix and provides detailed proofs and explanation for all theorems and lemmas. Furthermore, Theorems 4.2, 4.3, and 4.4, which relate the performance measure to known graph topologies, are new additions to this manuscript. Algorithm 2, which is superior to Algorithm 1, is also our new contribution. This superiority is shown through Example 9.5. In addition, materials in section VIII are new, and provide a theoretical guideline on which design methods should be utilized to enhance the performance.

In Section III, we express the $\mathcal{H}_2$-norm performance of a time-delay linear consensus network in terms of its Laplacian spectrum. Furthermore, we prove that this performance measure is convex with respect to coupling weights and Laplacian spectrum, and in addition, it is an increasing function of time-delay. In Section IV, we discuss topologies with optimal performance. Furthermore, we quantify a sharp lower bound on the best achievable performance for a network with a fixed time-delay. In the presence of time-delay, the $\mathcal{H}_2$-norm performance of a first-order consensus network is not monotone decreasing with respect to connectivity, which imposes challenges in the design of the optimal network as increasing connectivity may deteriorate the performance. Then, we present methods to improve the performance measure. We categorize these procedures as growing, reweighting, and sparsification. Although the $\mathcal{H}_2$-norm performance is a convex function of Laplacian eigenvalues, direct use of this spectral function in our network design problems requires eigendecomposition, which adds to time complexity of our design procedures. To overcome this, our key idea is to calculate an approximation function of the performance measure that spares us eigendecomposition of the Laplacian matrix. In Section V, we tackle the combinatorial problem of improving the nonmonotone performance measure of the time-delay network by adding new interconnection links. Our time-complexity analysis of our proposed algorithm to grow a time-delay network shows that it can be done in $\mathcal{O}(n^3 + mn^2 + kn^2)$ arithmetic operations, where n is the number of nodes, m is the number of rows of the output matrix C defined in Eq. (3b) below, and k is the maximum number of new interconnections. Section VI discusses reweighting of the coupling weights as an approach to improve the performance measure. This design problem can be cast as a semidefinite programming (SDP) problem, which inherits the time complexity of existing SDP solvers. In the absence of time-delay, removing interconnections will deteriorate the $\mathcal{H}_2$-norm performance due to monotonicity property of the performance measure. In Section VII, we explain how one can sparsify the coupling graph of the time-delay network to improve the performance measure. In Section IX, we discuss the sensitivity of the performance measure with respect to weight of couplings. This result helps us assess if performance is enhanced by growing or sparsifying network links, without the need to know the spectrum of Laplacian matrix. We discuss several case studies that demonstrate the the advantages of our proposed design algorithms over the existing ones in the literature.

II. Preliminaries and Definitions

A. Basic Definitions

Throughout this paper, the following notations will be used. We denote transpose and conjugate transpose of the matrix A by A^{T} and A^{H}, respectively. Also, the set of nonnegative (positive) real numbers is indicated by $\mathbb{R}_+$ ($\mathbb{R}_{++}$). An undirected weighted graph $\mathcal{G}$ is denoted by the triple $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (vertices) of the graph, $\mathcal{E}$ is the set of links (edges) of the graph, and $w : \mathcal{E} \to \mathbb{R}_{++}$ is the weight function that maps each link to a positive scalar. We let L to be the Laplacian of the graph, defined by $L = \Delta - A$, where Δ is the diagonal matrix of node degrees and A is the adjacency matrix of the graph. The $n \times 1$ vector of all zeros and ones are denoted by 0_n and $\mathbf{1}_n$, respectively, whereas $J_n = \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}$ is the $n \times n$ matrix of all ones. Furthermore, the $n \times n$ centering matrix is denoted by $M_n = I_n - \frac{1}{n} J_n$. For an undirected graph with n nodes, Laplacian eigenvalues are real and shown in an order sequence as $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. We denote the complete unweighted graph by $\mathcal{K}_n$. We indicate the Moore–Penrose pseudoinverse of a matrix P by $P^{\dagger} = [p^{\dagger}_{ij}]$ and we define $r_e(P) := p^{\dagger}_{ii} + p^{\dagger}_{jj} - 2p^{\dagger}_{ij}$ for every given link $e = \{i, j\}$. Accordingly, in a graph with a Laplacian matrix L, the effective resistance between two ends of a given link $e = \{i, j\}$ is denoted by $r_e(L)$. For a given n-tuple y, operator $\mathrm{diag}(y)$ maps y to an $n \times n$ diagonal matrix whose main diagonal elements are elements of y. For a matrix $P \in \mathbb{R}^{n\times m}$, the vectorization of P, denoted by $\mathrm{Vec}(P) \in \mathbb{R}^{nm}$, is a vector obtained by stacking up columns of the matrix P on top of one another. For a square matrix X, matrix functions $\cos(X)$ and $\sin(X)$ are defined as follows:

$$\cos(X) = \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k}}{(2k)!}, \ \sin(X) = \sum_{k=0}^{\infty} \frac{(-1)^k X^{2k+1}}{(2k+1)!}. \quad (1)$$

Definition 2.1: A function $g : \mathbb{R}^n \to \mathbb{R}$ is Schur-convex if for every doubly stochastic matrix $D \in \mathbb{R}^{n\times n}$ and all $x \in \mathbb{R}^n$, we have $g(Dx) \leq g(x)$.

B. Noisy Consensus Networks With Time-Delay

We consider the class of linear dynamical networks that consists of multiple agents with scalar state variables x_i and control inputs u_i whose dynamics evolve in time according to

$$\dot{x}_i(t) = u_i(t) + \xi_i(t)$$

for all $i = 1, \ldots, n$. The impact of an uncertain environment on each agent's dynamics is modeled by the exogenous noise input $\xi_i(t)$. We assume that every agent experiences a time-delay in accessing, computing, or sharing its own state information with itself and other neighboring agents. All time-delays are taken identical and equal to a nonnegative number τ. We apply the following feedback control law:

$$u_i(t) = \sum_{j=1}^{n} k_{ij}\big(x_j(t-\tau) - x_i(t-\tau)\big) \tag{2}$$

to every agent of this network. The resulting closed-loop control system will be a first-order linear consensus network, whose dynamics can be written in the following compact form:

$$\dot{x}(t) = -L\,x(t-\tau) + \xi(t) \tag{3a}$$

$$y(t) = C\,x(t) \tag{3b}$$

with $x(t) = 0$ for all $t \in [-\tau, 0)$ and $x(0) = x^0$, where $x^0 = [x_1^0, \ldots, x_n^0]^{\mathrm{T}}$ is the initial condition, $x = [x_1, \ldots, x_n]^{\mathrm{T}}$ is the state, $y = [y_1, \ldots, y_m]^{\mathrm{T}}$ is the output, and $\xi = [\xi_1, \ldots, \xi_n]^{\mathrm{T}}$ is the exogenous noise input of the network. It is assumed that $\xi(t)$ is a vector of independent Gaussian white noise processes with zero mean and identity covariance, i.e.,

$$\mathbb{E}\left[\xi(t_1)\xi^{\mathrm{T}}(t_2)\right] = I_n\,\delta(t_1 - t_2)$$

where $\delta(t)$ is the delta function. The state matrix of the network is a graph Laplacian that is defined by $L = [l_{ij}]$, where

$$l_{ij} := \begin{cases} -k_{ij} & \text{if } i \neq j \\ k_{i1} + \cdots + k_{in} & \text{if } i = j \end{cases}.$$

Assumption 2.2: The vector of all ones is in the null space of the output matrix, i.e., $C\,\mathbf{1}_n = \mathbf{0}$.

The underlying coupling graph of the consensus network (3a) and (3b) is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ with node set $\mathcal{V} = \{1, \ldots, n\}$, edge set

$$\mathcal{E} = \Big\{\{i,j\}\,\big|\,\forall\, i,j \in \mathcal{V},\ k_{ij} \neq 0\Big\}$$

and weight function $w(e) = k_{ij}$ for all $e = \{i,j\} \in \mathcal{E}$, and $w(e) = 0$ if $e \notin \mathcal{E}$. The Laplacian matrix of graph $\mathcal{G}$ is equal to L.

Assumption 2.3: Feedback gains $\{k_{ij}\}_{i,j\in\mathcal{V}}$ satisfy the properties of: (i) non-negativity $k_{ij} \geq 0$, (ii) symmetry $k_{ij} = k_{ji}$, and (iii) simpleness $k_{ii} = 0$.

Property (ii) implies that the underlying graph $\mathcal{G}$ is undirected and property (iii) means that there is no self loop in the network.

Assumption 2.4: The coupling graph $\mathcal{G}$ of the consensus network (3a) and (3b) is connected and time invariant.

This assumption implies that the smallest Laplacian eigenvalue is equal to zero, i.e., $\lambda_1 = 0$, while the remaining ones are strictly positive, i.e., $\lambda_i > 0$ for $i = 2, ..., n$.

C. Network Performance Measures

When there is no input noise, i.e., $\xi(t) \equiv 0_n$, it is already known [10] that under condition $\lambda_n < \frac{\pi}{2\tau}$ states of all agents converge to average of all initial states; whereas, in the presence of input noise, the agents' states fluctuate around around the moving average.

In order to quantify the quality of noise propagation in the dynamical network (3), we adopt the following performance measure:

$$\rho_{ss}(L;\tau) = \lim_{t\to\infty} \mathbb{E}\left[y^{\mathrm{T}}(t)y(t)\right]. \tag{4}$$

It can be verified that the transfer function of the consensus network (3) is equivalent to the transfer function of the following system:

$$\begin{cases} \dot{\hat{x}}(t) &= -\big(L + \frac{1}{\tau n}J_n\big)\hat{x}(t-\tau) + M_n\xi(t) \\ y(t) &= C\hat{x}(t) \end{cases} \tag{5}$$

where $\hat{x}(t) = M_n x(t)$ is the projection of network's states on to the disagreement subspace, i.e., $\hat{x}_i(t) = x_i(t) - \bar{x}(t)$ in which

$$\bar{x}(t) = \frac{1}{n}\big(x_1(t) + \cdots + x_n(t)\big).$$

Since for $\tau < \frac{\pi}{2\lambda_n}$, the system (5) is exponentially stable and the transfer function of system (5) is identical to the transfer function of system (3); we infer that the marginally stable mode of the consensus network (3) (which corresponds to $\lambda_1 = 0$) is not observable in the output $y(t)$ according to Assumption 2.2. This results in existence and boundedness of the performance measure (4). This method has been widely exploited in the literature [3], [4], [28], [29].

We now list three coherency measures that have recently been proposed in the context of a linear consensus network [2]–[4].

1) *Pairwise deviation*:

$$\frac{1}{2n}\sum_{i,j=1}^{n}(x_i - x_j)^2 = x^T(t)B_{\mathcal{K}_n}^{\mathrm{T}}\,\mathrm{diag}\left(\frac{1}{n}, \ldots, \frac{1}{n}\right)B_{\mathcal{K}_n}x(t) = x^T(t)M_n x(t)$$

where $B_{\mathcal{K}_n}$ is the signed edge-to-vertex incidence matrix of the complete graph $\mathcal{K}_n$.

2) *Deviation from average*:

$$\|x(t) - \bar{x}(t)\mathbf{1}_n\|_2^2 = \sum_{i=1}^{n}(x_i(t) - \bar{x}(t))^2 = x^T(t)M_n x(t).$$

3) *Norm of projection onto the stable subspace*: When there is no noise, network (3a) is marginally stable and we only consider the dynamics on the stable subspace of $\mathbb{R}^n$ that is orthogonal to the subspace spanned by $\mathbf{1}_n$. For a given $S \in \mathbb{R}^{(n-1)\times n}$ whose rows form an orthonormal basis of the disagreement subspace, the norm of the projection of $x(t)$ onto the stable subspace is a coherency measure [3] that is given by $\|Sx(t)\|_2^2 = x^T(t)SS^Tx(t)$ where $S\mathbf{1}_n = 0$ and $S^T S = M_n$.

It can be proven that our utilized measure of performance (4) is equal to the square of $\mathcal{H}_2$-norm of the network from ξ to y. Thus, we utilize the interpretation of energy of impulse response in order to calculate performance measure of the network [30], i.e., we have

$$\rho_{ss}(L;\tau) = \frac{1}{2\pi}\,\mathrm{Tr}\left[\int_{-\infty}^{+\infty} G^{\mathrm{H}}(j\omega)G(j\omega)\,d\omega\right] \tag{6}$$

where $G(s)$ is the transfer function of (3) from ξ to y.

D. Problem Statement

Our main objective is to explore all possible ways to improve performance of the time-delay first-order consensus networks (3) with respect to performance measure (4). In order to tackle this problem, first we need to quantify performance measure (4) in terms of the spectrum of the underlying coupling graph of the network. Next, we need to characterize inherent fundamental limits on the best achievable performance and classify those networks that can actually achieve this hard limit. There are only three possible ways to improve the performance of network (3) by manipulating its underlying coupling graph: growing, sparsification, and reweighting. Therefore, we need to investigate under what conditions performance can be improved in each of these three possible scenarios. Improving performance in the presence of time-delay is a challenging task due to the counterintuitive effects of connectivity on the performance.

III. Properties of the Performance in the Presence of Time-Delay

In the following theorem, we derive an exact expression for the performance measure of the consensus network.

Theorem 3.1: For the dynamical network (3), the performance measure (4) can be specified by

$$\rho_{ss}(L;\tau)=\frac{1}{2}\,\mathrm{Tr}\left[L_o L^{\dagger}\cos(\tau L)\big(M_n-\sin(\tau L)\big)^{\dagger}\right] \tag{7}$$

where $L_o=C^{\mathrm{T}}C$. In addition, when the output matrix is equal to the centering matrix, i.e., $C=M_n$, the performance measure can be quantified as an additively separable function of Laplacian eigenvalues; in other words, we have the following formula:

$$\rho_{ss}(L;\tau)=\sum_{i=2}^{n} f_\tau(\lambda_i) \tag{8}$$

where $f_\tau(\lambda_i)=\frac{1}{2\lambda_i}\frac{\cos(\lambda_i\tau)}{1-\sin(\lambda_i\tau)}$.

Proof: In order to find the performance of network (3), we utilize (6)

$$\rho_{ss}(L;\tau)=\frac{1}{2\pi}\,\mathrm{Tr}\left[\int_{-\infty}^{+\infty}G^{\mathrm{H}}(j\omega)G(j\omega)\,d\omega\right]$$

where $G(s)$ is the transfer function of both (3) and (5), i.e.,

$$G(s)=C\left(sI_n+e^{-\tau s}\left(L+\frac{1}{\tau n}J_n\right)\right)^{-1}M_n. \tag{9}$$

We consider the spectral decomposition of the Laplacian matrix L, which is given as follows: $L=Q\Lambda Q^T$ where $Q=[q_1,q_2,\dots,q_n]\in\mathbb{R}^{n\times n}$ is the orthonormal matrix of eigenvectors and $\Lambda=\mathrm{diag}(\lambda_1,\dots,\lambda_n)$ is the diagonal matrix of eigenvalues. We recall that $\lambda_1=0$ for the reason that the graph is undirected and it has no self loops. Therefore

$$\begin{aligned}M_n&=I_n-Q\,\mathrm{diag}(1,0,\dots,0)Q^T\\&=Q\,\mathrm{diag}(0,1,\dots,1)Q^T\end{aligned} \tag{10}$$

and

$$L=Q\,\mathrm{diag}(0,\lambda_2,\dots,\lambda_n)Q^T. \tag{11}$$

Thus

$$L+\frac{1}{\tau n}J_n=Q\,\mathrm{diag}\left(\frac{1}{\tau},\lambda_2,\dots,\lambda_n\right)Q^T \tag{12}$$

and substituting (10) and (12) into (9) we get

$$G(s)=CQ\,\mathrm{diag}\left(0,\frac{1}{s+\lambda_2 e^{-\tau s}},\dots,\frac{1}{s+\lambda_n e^{-\tau s}}\right)Q^T.$$

Hence, we have

$$\begin{aligned}&\mathrm{Tr}\left[G^{\mathrm{H}}(j\omega)G(j\omega)\right]\\&=\mathrm{Tr}\left[C^{\mathrm{T}}CQ\,\mathrm{diag}\left(0,\frac{1}{-j\omega+\lambda_2 e^{j\tau\omega}},\dots,\frac{1}{-j\omega+\lambda_n e^{j\tau\omega}}\right)\right.\\&\left.\mathrm{diag}\left(0,\frac{d\omega}{j\omega+\lambda_2 e^{-j\tau\omega}},\dots,\frac{1}{j\omega+\lambda_n e^{-j\tau\omega}}\right)Q^T\right]\end{aligned} \tag{13}$$

and by substituting (13) in the following definition of $\rho_{ss}(L;\tau)$:

$$\begin{aligned}\rho_{ss}(L;\tau)&=\frac{1}{2\pi}\,\mathrm{Tr}\left[\int_{-\infty}^{+\infty}G^{\mathrm{H}}(j\omega)G(j\omega)\,d\omega\right]\\&=\frac{1}{2\pi}\sum_{i=2}^{n}\int_{-\infty}^{+\infty}\frac{\Delta_i\quad d\omega}{(j\omega+\lambda_i e^{-j\tau\omega})(-j\omega+\lambda_i e^{j\tau\omega})}\end{aligned} \tag{14}$$

where Δ_i is the ith diagonal element of the matrix $Q^{\mathrm{T}}L_oQ$. By applying Lemma 10.7 of [31] to (14), we get

$$\rho_{ss}(L;\tau)=\sum_{i=2}^{n}\frac{\Delta_i}{2\lambda_i}\,\frac{\cos(\lambda_i\tau)}{1-\sin(\lambda_i\tau)}. \tag{15}$$

From eigenvalue decomposition (11), definition (1), simultaneous diagonalizability of L, $\cos(\tau L)$ and $\big(M_n-\sin(\tau L)\big)$, and notation, we define the function of the Laplacian matrix as follows:

$$\begin{aligned}f_\tau(L)&=Q\,\mathrm{diag}\big(0,\ f_\tau(\lambda_2),\dots,\ f_\tau(\lambda_n)\big)\,Q^T\\&=\frac{1}{2}L^{\dagger}\cos(\tau L)\Big(M_n-\sin(\tau L)\Big)^{\dagger}.\end{aligned}$$

Now, we can rewrite equality (15) in the following compact matrix operator form:

$$\rho_{ss}(L;\tau)=\frac{1}{2}\,\mathrm{Tr}\left[L_oL^{\dagger}\cos(\tau L)\Big(M_n-\sin(\tau L)\Big)^{\dagger}\right].$$

In addition, when $C=M_n$, we have

$$\rho_{ss}(L;\tau)=\frac{1}{2}\,\mathrm{Tr}\left[L^{\dagger}\cos(\tau L)\Big(M_n-\sin(\tau L)\Big)^{\dagger}\right]$$

and $\Delta_i=1$ for $i\geq 2$, from which one can deduct (8). ■

Remark 3.2: This result for the case that $C=M_n$ was derived in [20] and [32] using complex analysis techniques. We obtained this independently with a state-space point view for a general output matrix C that is orthogonal to vector of all ones.

Remark 3.3: Simultaneous diagonalizability of $\cos(\tau L)$, $\sin(\tau L)$, and L follows from (1) and the fact that L is diagonalizable. Also, as we showed in the proof of the theorem above, M_n and L share same set of eigenvectors and thus they are simultaneous diagonalizable. Hence, $M_n-\sin(\tau L)$ and L are simultaneous diagonalizable as well.

When there is no time-delay in the network, i.e., $\tau=0$, our result reduces to those of [3], [1], and [33], in which

$$\rho_{ss}(L;0)=\frac{1}{2}\sum_{i=2}^{n}\lambda_i^{-1}.$$

Theorem 3.4: The performance measure (4) for the consensus network (3) with a fixed underlying graph is an increasing function of the time-delay, i.e., the following inequality holds:

$$\rho_{ss}(L;\tau_1)<\rho_{ss}(L;\tau_2)$$

for every $0\leq\tau_1<\tau_2<\frac{\pi}{2\lambda_n}$.

Proof: As a means to demonstrate that $\rho_{ss}(L;\tau)$ is increasing in τ, we show that in the stability region, first derivative of performance measure with respect to τ is positive. Following this idea, we get

$$\frac{d}{d\tau}\rho_{ss}(L;\tau) = \frac{1}{2}\,\mathrm{Tr}\Big[-L_o\sin(\tau L)\big(M_n-\sin(\tau L)\big)^{\dagger} + L_o\cos^2(\tau L)\Big(\big(M_n-\sin(\tau L)\big)^{\dagger}\Big)^2\Big]$$

for all $\tau \in (0, \frac{\pi}{2\lambda_n})$. Due to simultaneous diagonalizability of $L, \sin(\tau L), \cos(\tau L)$, and $\big(M_n - \sin(\tau L)\big)$, their product is commutative, and therefore, we have

$$\frac{d}{d\tau}\rho_{ss}(L;\tau) = \frac{1}{2}\,\mathrm{Tr}\left[L_o\big(M_n-\sin(\tau L)\big)^{\dagger}\right] = \frac{1}{2}\,\mathrm{Tr}\left[C\big(I_n-\sin(\tau L)\big)^{-1}C^T\right]$$

where the last equality follows from spectral properties of L and orthogonality of rows of matrix C with respect to 1_n. Since matrix $\big(I_n - \sin(\tau L)\big)^{-1}$ is positive definite and C has nonzero components, we have

$$\frac{d}{d\tau}\rho_{ss}(L;\tau) > 0.$$

∎

Theorem 3.5: For a fixed time-delay, if $C \in \mathbb{R}^{(n-1)\times n}$ such that rows of C span the disagreement subspace or $C = B_{\mathcal{K}_n}$, the performance measure (4) for the dynamical network (3) is a convex and Schur-convex function of Laplacian eigenvalues of its underlying graph. In addition, the performance measure is a convex function of weight of links of the underlying graph $\mathcal{G}$.

Proof: For all $\lambda \in (0, \frac{\pi}{2\tau})$, the following inequality holds:

$$\sin\left(\tau\lambda + \frac{\pi}{4}\right) > \frac{\sqrt{2}}{2}.$$

By expanding the left-hand side of the above inequality, we get

$$\sin(\tau\lambda) + \cos(\tau\lambda) - 1 > 0. \tag{16}$$

Moreover, by multiplying both sides of (16) by $\tau\lambda$, we have

$$\tau\lambda\sin(\tau\lambda) - \tau\lambda + \tau\lambda\cos(\tau\lambda) > 0.$$

Since $\tau\lambda < 2$, it follows that

$$\tau\lambda\sin(\tau\lambda) - \tau\lambda + 2\cos(\tau\lambda) > 0.$$

Subsequently, we get the following inequality by multiplying both sides by $(\sin(\tau\lambda) - 1)$ and $\sec^3(\tau\lambda)$, which are, respectively, negative and positive

$$(\sin(\tau\lambda)-1)\sec^3(\tau\lambda)(\tau\lambda(\sin(\tau\lambda)-1)+2\cos(\tau\lambda)) < 0.$$

The left-hand side of the above inequality equals to $\frac{d^2}{d\lambda^2}(\lambda(\sec(\tau\lambda) + \tan(\tau\lambda)))$; thus, we deduce that

$$\frac{d^2}{d\lambda^2}\left(\frac{1}{f_\tau(\lambda)}\right) < 0. \tag{17}$$

Moreover, from inequality (17), it follows that

$$f_\tau(\lambda)\frac{d^2}{d\lambda^2}f_\tau(\lambda) > 0.$$

Consequently, positiveness of $f_\tau(\lambda)$ results in strict convexity of $f_\tau(\lambda)$. Since the performance function $\rho_{ss}(L;\tau)$ equals to sum of convex functions, it is a convex function of Laplacian eigenvalues. Since $\rho_{ss}(L;\tau)$ is a convex function of the eigenvalues of the Laplacian matrix and it is a symmetric function of the eigenvalues, it is a Schur-convex function Theorem 10.6.

Applying Davis's theorem [34], since the network performance is a symmetric convex function of eigenvalues, it is also a convex function of the Laplacian matrix. Therefore, performance measure is a convex function of weight of links of the coupling graph. ∎

Convexity and Schur-convexity properties of the performance function helps us to find fundamental limits on the best achievable performance as well as upper bounds on the performance measure of the network without knowing the spectrum of the Laplacian matrix of the underlying graph [35].

IV. Optimal and Robust Topologies W.R.T. Time-Delay

The following result characterizes the optimal interconnection topology for a consensus network in presence of time-delay.

Theorem 4.1: For the first-order linear consensus network (3) with n nodes that is affected by time-delay τ, the limit on the best achievable performance is given by

$$\rho_{ss}(L;\tau) \geq \frac{\tau\|C\|_F^2}{2\big(1-\sin(z^*)\big)} \tag{18}$$

where $z^* > 0$ is the unique positive solution of $\cos(z) = z$. Furthermore, the optimal topology in terms of the performance measure is a complete graph with identical weight

$$w^*(e) = \frac{z^*}{n\tau} \tag{19}$$

for every coupling link e.

Proof: For a fixed time-delay, performance measure will be minimized if $f_\tau(\lambda_i)$ is minimized for all λ_i, where $i \in \{2,\ldots,n\}$. By strict convexity and twice differentiability of $f_\tau(\lambda_i)$, minimum of performance measure is attained if and only if

$$\lambda_i = \arg\min_{\lambda_i} f_\tau(\lambda_i) = \left\{\lambda_i \,\middle|\, \frac{d}{d\lambda_i}f_\tau(\lambda_i) = 0\right\}.$$

for $i \in \{2,\ldots,n\}$. Solving $\frac{d}{d\lambda_i}f_\tau(\lambda_i) = 0$ results in the following equation:

$$\cos(\lambda_i\tau) = \lambda_i\tau.$$

Since z^* is the solution of the equation $\cos(z) = z$, we have

$$\lambda_i = \frac{z^*}{\tau} \tag{20}$$

for all $i \in \{2,\ldots,n\}$. In addition, substituting λ_i from the previous equation in (15), we obtain

$$\rho_{ss}(L;\tau) \geq \frac{\tau\|C\|_F^2}{2\big(1-\sin(z^*)\big)}\sum_{i=2}^{n}\Delta_i.$$

Consequently, since $\sum_{i=2}^{n}\Delta_i = \mathrm{Tr}[Q^{\mathrm{T}}C^{\mathrm{T}}CQ]$, the fundamental limit given by inequality (18) can be deduced. Moreover, considering equality of all eigenvalues of L, the underlying graph of the network with the optimal performance is a complete graph with equal link weights. Besides, since nonzero eigenvalues of a complete graph with uniform link weights $w^*(e)$, for all links e, are

$$\lambda_i = nw^*(e) \tag{21}$$

for all $i \in \{2, \ldots, n\}$, substituting λ_i from (21) into (20) yields identity (19). ■

The lower bound of the coherency for the case that $C = M_n$ was found using the numerical analysis in [20] and [21]. We found the fundamental limit for the general output matrix C and studied uniqueness of the limit using convex analysis.

When $\tau = 0$, it is known that the best achievable performance for linear consensus networks with weighted underlying graphs can be made arbitrarily small [1]. This is consistent with the result of Theorem 4.1 since the best achievable performance over all possible network topologies approaches zero as time-delay goes to zero. It is noteworthy that for the first-order linear consensus networks (4), the best attainable performance grows linearly with time-delay. Also, under assumption of fixed time-delay, the best achievable performance increases linearly with network size, i.e., it is in the order of $\mathcal{O}(n)$. Furthermore, weight of the links in the network with optimal performance is inversely proportional to network size. In the following theorems, we classify graph topologies of robust consensus networks with respect to time-delay increments. We also note that if rows of the output matrix C span the disagreement subspace, then the optimal topology would be unique.

Theorem 4.2: Suppose that L_1 and L_2 are Laplacian matrices of coupling graphs of two consensus networks governed by (4). If $\lambda_n(L_1) > \lambda_n(L_2)$ and rows of the output matrix C span the disagreement subspace, then there exists a threshold $\tau^* > 0$ such that for all $\tau > \tau^*$ the following ordering holds:

$$\rho_{ss}(L_2;\tau) < \rho_{ss}(L_1;\tau).$$

Moreover, the value of τ^* depend on L_1 and L_2 and the output matrix.

Proof: As τ approaches $\frac{\pi}{2\lambda_n(L_1)}$ from left, $\rho_{ss}(L_1;\tau)$ increases unboundedly. A proof follows from boundedness of $\rho_{ss}(L_2;\tau)$ for $\tau \in [0, \frac{\pi}{2\lambda_n(L_1)})$ and unboundedness of $\frac{\pi}{2\lambda_n(L_1)}$ in the same interval. The smallest τ^* is the solution of a nonlinear equation although in the following we show that any

$$\tau \in \left[\frac{\pi\hat{p}}{2\hat{p}\lambda_n(L_1)+1}, \frac{\pi}{2\lambda_n(L_1)}\right) \tag{22}$$

can serve as a τ^*, where $\hat{p} = \rho_{ss}(L_2; \frac{\pi}{2\lambda_n(L_1)})$. Based on the definition of ρ_{ss}, we have

$$\rho_{ss}(L_1;\tau) \geq \frac{\Delta_n}{2\lambda_n(L_1)} \frac{\cos\left(\tau\lambda_n(L_1)\right)}{1-\sin\left(\tau\lambda_n(L_1)\right)}$$

for all $\tau \in [0, \frac{\pi}{2\lambda_n(L_1)})$. Furthermore, since $\frac{1}{2\lambda_n(L_1)} > \frac{\tau}{\pi}$, we have

$$\rho_{ss}(L_1;\tau) \geq \frac{\Delta_n \tau}{\pi} \frac{\pi/2}{\pi/2 - \tau\lambda_n(L_1)}.$$

Using the above inequality, if condition (22) holds, then

$$\rho_{ss}(L_1;\tau) \geq \rho_{ss}\left(L_2; \frac{\pi}{2\lambda_n(L_1)}\right).$$

Thus, our desired result follows from the result of Theorem 3.4. ■

Theorem 4.3: Suppose that two linear consensus networks with dynamics (3) and unweighted underlying graphs are given such that, the graph topology of one of them is path, denoted by $\mathcal{P}$, and the other one has an arbitrary nonpath graph topology, shown by $\mathcal{G}$. If rows of the output matrix C span the disagreement subspace, then there exists a $\tau^*_{\mathcal{G}} > 0$ such that for all $\tau > \tau^*_{\mathcal{G}}$, the following ordering holds:

$$\rho_{ss}(L_{\mathcal{G}};\tau) > \rho_{ss}(L_{\mathcal{P}};\tau).$$

Proof: For a path graph on n nodes, we have

$$\lambda_n(L_{\mathcal{P}}) = 2 - 2\cos\left(\frac{n-1}{n}\pi\right) < 4. \tag{23}$$

Whereas, for any nonpath topology $\mathcal{G}$ and $n > 3$, we have $d_{\max}(\mathcal{G}) \geq 3$. Thus, for any nonpath $\mathcal{G}$ with more than three nodes, by Lemma 10.8 of [31] and (23), we get $\lambda_n(L_{\mathcal{P}}) < 4 < \lambda_n(L_{\mathcal{G}})$. For graphs with more than three nodes, a proof follows from Theorem 4.2. We note that for $n = 3$ there exists only two topologies, namely path graph and complete graph. Even though, the largest eigenvalue of both of these graphs is equal to 3, the multiplicity of this eigenvalue for the complete graph is 2, and the required result follows. ■

Theorem 4.4: Suppose that two linear consensus networks with dynamics (3) and unweighted underlying graphs are given: one with ring topology $\mathcal{R}$ and the other one with an arbitrary non-ring topology $\mathcal{G}$ that has at least one loop. If $C = M_n$, then there exists a $\tau^*_{\mathcal{G}} > 0$ such that for all $\tau > \tau^*_{\mathcal{G}}$, the following inequality holds $\rho_{ss}(L_{\mathcal{G}};\tau) > \rho_{ss}(L_{\mathcal{R}};\tau)$.

Proof: For ring graphs with $n > 4$ nodes, we have $\lambda_n(L_{\mathcal{R}}) \leq 2 - 2\cos(\pi) = 4$. Whereas, for other non-ring, non-tree graph topologies $\mathcal{G}$ with more than four nodes, Lemma 10.8 of [31] yields

$$\lambda_n(L_{\mathcal{R}}) \leq 4 < \lambda_n(L_{\mathcal{G}}).$$

For $n > 4$, the proof follows from Theorem 4.2. For $n = 4$, our claim holds true since for all non-tree topologies, we have $\lambda_n = 4$, but the second largest eigenvalue for a ring graph is 2 and for the rest of non-tree topologies, it is greater than or equal to 3. From this, the desired result follows. ■

The results of Theorems 4.2 and 4.4 are in contrast with the common intuition of performance in nondelayed first-order linear consensus networks, where enhancing connectivity in the sense of $L_1 \preceq L_2$ always results in performance improvement, i.e., $\rho_{ss}(L_2;0) \leq \rho_{ss}(L_1;0)$ and path graph has the worst performance among all unweighted graphs. In conclusion, in time-delay linear consensus networks, higher connectivity does not necessarily imply better $\mathcal{H}_2$-norm performance [25].

In the subsequent sections, we exploit properties of ρ_{ss} to develop design algorithms that enhance systemic performance of the network. Furthermore, we discuss efficacy of these algorithms by comparing them with fundamental limits that we derived in this section.

V. Improving Performance by Adding New Feedback Gains

In Sections VI– VIII, we consider the problem of performance improvement in a time-delay linear consensus network. There are only three possible ways to achieve this objective via manipulating the underlying graph of the network: by adding new interconnection links (growing); by adjusting weight of existing links; and by eliminating existing links (sparsification). Other design objectives, such as rewiring, can be equivalently executed in several consecutive design steps involving reweighing, growing, and sparsification.

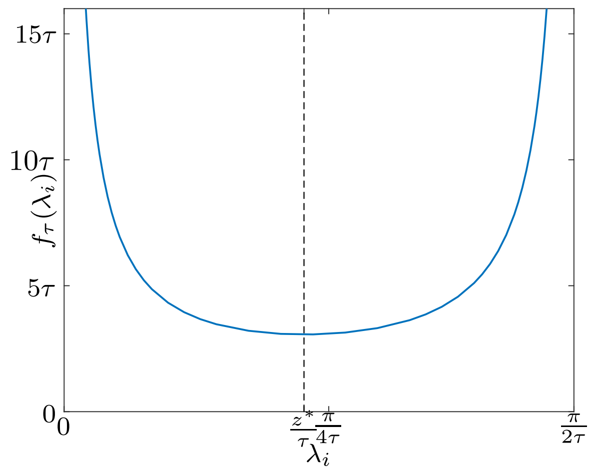

Fig. 1. This plot illustrates convexity property of $\rho_{ss}(L;\tau)$ by depicting $f_\tau(\lambda_i)$ as a function of Laplacian eigenvalues.

In this section, we consider the problem of growing consensus network (4), where it is allowed to establish new interconnections links in the network. It is assumed that some of the Laplacian eigenvalues are located on the left side of the dashed line in Fig. 1, i.e., $\lambda_i < \frac{z^*}{\tau}$ for some $i \in \{2, \ldots, n\}$. In this case, enhancing the connectivity can improve $\mathcal{H}_2$-norm performance of the network.

Suppose that a set of candidate links $\mathcal{E}_c$ and a corresponding weight function $\varpi : \mathcal{E}_c \to \mathbb{R}_+$ are given. Adding a new link between two agents is equivalent to closing a new feedback loop around these two agents according to our earlier interpretation (2). Therefore, weight of a candidate link plays role of a feedback gain in the overall closed-loop system and it cannot be chosen arbitrary; its value is opted by considering all existing constraints. Based on this elucidation, it is reasonable to consider the following modified form of network (3) for our design purpose:

$$\begin{aligned}\dot{x}(t) &= -Lx(t-\tau) + u(t) + \xi(t)\\ u(t) &= -L_{_F}\, x(t-\tau)\\ y(t) &= C\,x(t)\end{aligned}$$

that can be rewritten in the following closed-loop form:

$$\dot{x}(t) = -(L + L_{_F})x(t-\tau) + \xi(t) \tag{24a}$$

$$y(t) = C\,x(t) \tag{24b}$$

where $L_{_F}$ is the Laplacian matrix of the feedback gain and can be represented as follows:

$$L_{_F} = \sum_{e\,\in\,\mathcal{E}_s} \varpi(e) b_e b_e^T$$

where b_e is the corresponding column to edge e in the node-to-edge incidence matrix of the underlying graph of the network. Our design objective is to improve performance of the noisy network in the presence of time-delay by designing a sparse Laplacian feedback gain $L_{_F}$ with at most k links having predetermined weight, i.e., our goal is to solve the following optimization problem:

$$\underset{\mathcal{E}_s}{\text{minimize}} \quad \rho_{ss}(L + L_{_F};\tau) \tag{25}$$

$$\text{subject to:} \quad L_{_F} = \sum_{e\in\mathcal{E}_s} \varpi(e) b_e b_e^{\mathrm{T}} \tag{26}$$

$$0 \preceq L + L_{_F} \prec \frac{\pi}{2\tau} I_n \tag{27}$$

$$|\mathcal{E}_s| \leq k \qquad \text{for all} \quad \mathcal{E}_s \subseteq \mathcal{E}_c. \tag{28}$$

Condition (27) ensures stability of the closed-loop network (27). Since the problem given by (25)–(28) is combinatorial, the exact solution must be found by an exhaustive search and appraising $\rho_{ss}(L + L_{_F};\tau)$ for all possible $\sum_{i=1}^{k} \binom{|\mathcal{E}_c|}{i}$ cases. In real-world problems, when the size of candidate set is prohibitively large, we need efficient methods to tackle the problem. Furthermore, when there is no time delay, the $\mathcal{H}_2$-norm performance of the network will improve no matter how we choose and add the new candidate links [7]. However, in the presence of time delay, adding new links may deteriorate performance or even destabilize the closed-loop network, which is why growing a time-delayed network is a more delicate task.

A. Cost Function Approximation and SDP Relaxation

We can derive a convex relaxation of our problem by letting constants $\varpi(e)$ to become decision variables, shown by $w(e)$, and replacing the constraint (28) by

$$\sum_{e\in\mathcal{E}_c} w(e) \;\leq\; W_k \tag{29}$$

where

$$W_k = \max_{\substack{|\mathcal{E}_s|=k\\ \mathcal{E}_s\subseteq\mathcal{E}_c}} \sum_{e\in\mathcal{E}_s} \varpi(e). \tag{30}$$

Thus, our design optimization problem is to solve

$$\underset{\{w(e)|\forall e\in\mathcal{E}_c\}}{\text{minimize}} \quad \rho_{ss}(L + L_{_F};\tau)$$

$$\text{subject to:} \quad (29), (30), (32), (33). \tag{31}$$

In spite of smoothness of the cost function in (31), the structure of the cost function is not appealing since we cannot cast it as an SDP with linear objective function and constraints or solve it using existing and standard solvers or toolboxes. Moreover, if we want to write a solver for the problem using the conventional methods, e.g., interior-point or subgradient methods, we have to find eigenvalues and eigenvectors of $L + L_{_F}$ for each step of minimizing the performance function; which significantly increases complexity in terms of both time and details of solver. Therefore, we need an alternative way to remove eigendecomposition from our solution. We overcome this obstacle by introducing a tight approximation of (7) that has a small relative error with respect to our performance measure.

Lemma 5.1: For a stable linear consensus network (27) with the output matrix C with property $C\mathbf{1}_n = \mathbf{0}_n$, the spectral function

$$\tilde{\rho}_{ss}(L;\tau) = \frac{1}{2}\,\mathrm{Tr}\left[L_o L^\dagger + \frac{4\tau}{\pi} L_o \left(\frac{\pi}{2} M_n - \tau L\right)^\dagger + c_1\tau^2 L_o L + \frac{c_0}{2}\tau L_o\right]$$

approximates (7) with guaranteed error bound

$$0 \le \frac{\rho_{ss}(L;\tau) - \tilde{\rho}_{ss}(L;\tau)}{\rho_{ss}(L;\tau)} \le 2 \times 10^{-4}$$

where $c_0 = 0.18733$ and $c_1 = -0.01$ are constants to minimize the mean-squared error numerically.

Proof: We recall that

$$\rho_{ss}(L;\tau) = \sum_{i=2}^{n} \frac{\Delta_i}{2\lambda_i} \frac{\cos(\tau\lambda_i)}{1-\sin(\tau\lambda_i)}$$

by multiplying the nominator and denominator by τ we get

$$\begin{aligned}\rho_{ss}(L;\tau) &= \tau \sum_{i=2}^{n} \frac{\Delta_i}{2\tau\lambda_i} \frac{\cos(\tau\lambda_i)}{1-\sin(\tau\lambda_i)} \\ &= \tau \sum_{i=2}^{n} \Delta_i f_1(\tau\lambda_i) \qquad (32)\end{aligned}$$

where $f_1(x) = \frac{1}{2x}\frac{\cos(x)}{1-\sin(x)}$ with domain $x \in (0, \pi/2)$ based on definition of f_τ. As a means to find a proper approximate performance function, we look for an approximation of f_1 and we denote it by $\tilde{f}$. Since f_1 has two vertical asymptotes inside its effective domain, we want to have bounded $\|\tilde{f} - f_1\|_\infty$ over effective domain of these functions. To that end, we utilize

$$\tilde{f}(x) = \frac{1}{2}\left(\frac{1}{x} + \frac{4}{\pi}\frac{1}{\pi/2 - x} + c_0 + c_1 x\right) \qquad (33)$$

as approximation of f_1 where $c_0 = 0.18733$ and $c_1 = -0.01$ are constants to minimize the mean-squared error numerically. We obtain our performance approximate function by substituting f_1 with $\tilde{f}$ (32). Thus, from relative error of $\tilde{f}$ with respect to f_1 given in Fig. 2, it yields the desired approximation bound. Consequently, we can write the approximation function in the following form:

$$\begin{aligned}\tilde{\rho}_{ss}(L;\tau) &= \tau \sum_{i=2}^{n} \Delta_i \tilde{f}(\tau\lambda_i) \\ &= \tau \sum_{i=2}^{n} \frac{\Delta_i}{2}\left(\frac{1}{\tau\lambda_i} + \frac{4}{\pi}\frac{1}{\pi/2 - \tau\lambda_i} + c_0 + c_1\tau\lambda_i\right) \\ &= \frac{1}{2}\operatorname{Tr}\left[L_o L^\dagger + \frac{4\tau}{\pi} L_o \left(\frac{\pi}{2} M_n - \tau L\right)^\dagger + c_1\tau^2 L_o L + \frac{c_0}{2}\tau L_o\right].\end{aligned}$$

Remark 5.2: Performance of a first-order consensus network with $C = M_n$ was previously approximated in [21] and [36] by using heuristic methods. We have utilized a systematic method to provide an approximate function (33) and we refer to [37] for more details.

It is straightforward to show that $\tilde{\rho}_{ss}$ is convex function of eigenvalues and weights of the links for any C such that $C\mathbf{1}_n = \mathbf{0}_n$. Replacing ρ_{ss} by $\tilde{\rho}_{ss}$ and combinatorial constraint (28) by its relaxed form (29), we can relax (31) to the following optimization problem:

$$\begin{aligned}&\underset{\{w(e)|\forall e \in \mathcal{E}_c\}}{\text{minimize}} \quad \tilde{\rho}_{ss}(L + L_F;\tau) \\ &\text{subject to:} \quad (26), (27), (29), (30). \qquad (34)\end{aligned}$$

In addition, neglecting the constant term in $\tilde{\rho}_{ss}$, the optimization problem (34) is equivalent to the following SDP:

$$\underset{\{w(e)|\forall e \in \mathcal{E}_c\}}{\text{minimize}} \quad \operatorname{Tr}\left[L_o X_1 + \frac{4}{\pi} L_o X_2 + c_1 \tau L_o L_F\right] \qquad (35)$$

$$\text{subject to:} \quad L_F = \sum_{e \in \mathcal{E}_c} w(e) b_e b_e^T \qquad (36)$$

$$\sum_{e \in \mathcal{E}_c} w(e) \le W_k \qquad (37)$$

$$\begin{bmatrix} X_1 & I \\ I & \tau(L + L_F) + \frac{1}{n}J \end{bmatrix} \succeq 0 \qquad (38)$$

$$\begin{bmatrix} X_2 & I \\ I & \frac{\pi}{2} I_n - \tau(L + L_F) \end{bmatrix} \succeq 0. \qquad (39)$$

The following theorem investigates the efficacy of using $\tilde{\rho}_{ss}$ as in an optimization problem (35)–(39).

Theorem 5.3: Let L_F^* be the solution of the optimization problem (31) and $\hat{L}_F$ be the solution of the minimization problem (34) where $C = M_n$ or C is an orthonormal matrix Q such that $Q\mathbf{1}_n = \mathbf{0}_n$. Then, we have

$$\rho_{ss}(L + L_F^*;\tau) \le \rho_{ss}(L + \hat{L}_F;\tau) \le (1+\epsilon)\rho_{ss}(L + L_F^*;\tau)$$

where $\epsilon = 2 \times 10^{-4}$.

Proof: First, we look at relative error of f_1 with $\tilde{f}$. For $\epsilon_1 = 2 \times 10^{-4}$, we have

$$0 \le \frac{f_1(\lambda_i) - \tilde{f}(\lambda_i)}{f_1(\lambda_i)} \le \epsilon_1.$$

In addition, rearranging and doing summation over all λ_is yield

$$\rho_{ss}(L;\tau) \le \tilde{\rho}_{ss}(L;\tau) \le (1+\epsilon_1)\rho_{ss}(L;\tau). \qquad (40)$$

Moreover, since $L + \hat{L}_F$ and $L + L_F^*$ are minimizers of $\tilde{\rho}_{ss}$ and ρ_{ss}, respectively, we observe that

$$\tilde{\rho}_{ss}(L + \hat{L}_F;\tau) \le \tilde{\rho}_{ss}(L + L_F^*;\tau) \qquad (41)$$

$$\rho_{ss}(L + L_F^*;\tau) \le \rho_{ss}(L + \hat{L}_F;\tau). \qquad (42)$$

Inequalities (40) and (41) yield that

$$\rho_{ss}(L + \hat{L}_F;\tau) \le (1+\epsilon_1)\rho_{ss}(L + L_F^*;\tau). \qquad (43)$$

Finally, from (42) and (43), we obtain

$$\rho_{ss}(L + L_F^*;\tau) \le \rho_{ss}(L + \hat{L}_F;\tau) \le (1+\epsilon_1)\rho_{ss}(L + L_F^*;\tau)$$

for $\epsilon_1 = 2 \times 10^{-4}$. ∎

B. Greedy Algorithms

In spite of the fact that the SDP relaxation of our problem can be solved using conventional SDP solvers, it cannot be utilized to improve performance of a moderately sized network (more than 20 000 candidate edges) as it would require a large amount of memory, which is not practically plausible. To address this issue, and in light of Theorem 5.3, we propose greedy algorithms to tackle the optimal control problem given in (25)–(28) for moderately sized networks. An undesirable naive procedure for one step of greedy algorithm is to choose the optimal link by evaluating the performance measure after adding the candidate links to the network one at a time, which involves computing the pseudoinverse of the Laplacian matrix for each candidate link. A positive aspect of using $\tilde{\rho}_{ss}$ as performance function is that it spares us the complexity of using eigendecompsition for the Laplacian matrix. The following theorem highlights an additional positive aspect of utilizing $\tilde{\rho}_{ss}$ instead of ρ_{ss} that enables us to calculate a useful explicit rank-one update rule.

Theorem 5.4: Let L_e be the rank-one weighted Laplacian matrix of a graph with only a single edge e between nodes i and j nodes with a given weight $\varpi(e)$. Then

$$\tilde{\rho}_{ss}(L+L_e;\tau) = \tilde{\rho}_{ss}(L;\tau) + c(e) \tag{44}$$

where

$$\begin{aligned} c(e) := &-\frac{r_e(LL_o^\dagger L)}{2\varpi(e)^{-1} + 2r_e(L)} + c_1\tau^2\varpi(e) \\ &- \frac{2\tau}{\pi}\frac{r_e\big((\frac{\pi}{2}M_n - \tau L)L_o^\dagger(\frac{\pi}{2}M_n - \tau L)\big)}{-(\varpi(e)\tau)^{-1} + r_e(\frac{\pi}{2}M_n - \tau L)}. \end{aligned} \tag{45}$$

Proof: Rearranging (44) yields

$$\begin{aligned} c(e) &= \tilde{\rho}_{ss}(L+L_e;\tau) - \tilde{\rho}_{ss}(L;\tau) \\ &= \frac{1}{2}\operatorname{Tr}[L_o(L+L_e)^\dagger - L_oL^\dagger] \\ &\quad + \frac{\tau}{2}\operatorname{Tr}[c_1\tau L_o(L+L_e) - c_1\tau L_oL] \\ &\quad + \frac{2\tau}{\pi}\operatorname{Tr}\left[L_o\left(\frac{\pi}{2}M_n - \tau(L+L_e)\right)^\dagger\right] \\ &\quad - \operatorname{Tr}\left[L_o\left(\frac{\pi}{2}M_n - \tau L\right)^\dagger\right]. \end{aligned} \tag{46}$$

Since L_e is a rank-one Laplacian matrix of a graph with a single edge between nodes i and j with weight $\varpi(e)$, we have

$$L_e = \varpi(e)(\chi_i - \chi_j)(\chi_i - \chi_j)^T$$

and further utilizing the Sherman–Morrison formula [38] for rank-one update, we have

$$\begin{aligned} \operatorname{Tr}\left[L_o(L+L_e)^\dagger - L_oL^\dagger\right] = \operatorname{Tr}\Big[L_oL^\dagger \\ - \frac{\varpi(e)L_oL^\dagger(\chi_i-\chi_j)(\chi_i-\chi_j)^{\mathrm{T}}L^\dagger}{1+\varpi(e)(\chi_i-\chi_j)^{\mathrm{T}}L^\dagger(\chi_i-\chi_j)} - L_oL^\dagger\Big]. \end{aligned}$$

In addition, using the cyclic permutation property of the trace operator, it yields that

$$\operatorname{Tr}\left[L_o(L+L_e)^\dagger - L_oL^\dagger\right] = -\frac{r_e(LL_o^\dagger L)}{\varpi(e)^{-1} + r_e(L)}.$$

Similarly, applying the Sherman–Morrison formula and the cyclic permutation for the trace to other terms of (46), (45) can be obtained. ■

If we let $\tau = 0$ and $L_o = M_n$ in (44), we obtain

$$\tilde{\rho}_{ss}(L+L_e;0) = \tilde{\rho}_{ss}(L;0) - \frac{r_e(L^2)}{2\varpi(e)^{-1} + 2r_e(L)}$$

which is the contribution of a new edge on the $\mathcal{H}_2$ performance when there is no time-delay [7], [8].

Although $\mathcal{H}_2$-performance measure of a consensus network (3) is not monotone in general with respect to adding new interconnection links to the coupling graph of the network [25], we can guarantee monotonicity of the $\mathcal{H}_2$-norm by imposing an upper bound on time-delay. More precisely, let us denote by Δ_d the maximum possible node degree among all the graphs over the set of all candidate augmented graphs; these are graphs that are obtained by adding k edges from candidate set $\mathcal{E}_c$ to the original graph for all possible choices.

Lemma 5.5: In the linear consensus network (4), if time delay satisfies $\tau < \frac{z^*}{2\Delta_d}$, then performance measures ρ_{ss} and $\tilde{\rho}_{ss}$, given in Theorem 3.1 and Lemma 5.1, respectively, will be monotone functions of the Laplacian matrix of the network, where z^* is the positive solution of $\cos(z) = z$.

If the performance measure is not monotone, one needs to verify whether adding new interconnection links destabilizes the network.

Theorem 5.6: Adding a new link e with weight $\varpi(e)$ to network (27) will retain stability of the network if and only if

$$\varpi(e) < w_s(L;e) \tag{47}$$

where $w_s(L;e) = (\tau\, r_e(\frac{\pi}{2}M_n - \tau L))^{-1}$.

Proof: We first show that if condition (47) does not hold, the network becomes unstable. When $\varpi(e)$ approaches $(\tau\ r_e(\frac{\pi}{2}M_n - \tau L))^{-1}$ from left, the denominator of the last term in the contribution of new edge to the performance (45) approaches infinity. Consequently, due to boundedness of the approximation function's error with respect to the performance function, as $w(e)$ approaches $(\tau\, r_e(\frac{\pi}{2}M_n - \tau L))^{-1}$ from left, the performance function goes to infinity and therefore the network goes to the verge of instability. On the other hand, if condition (47) holds, the contribution of a new edge to the performance will be bounded and therefore the performance will stay bounded and thus the system will remain stable. ■

According to Theorem 5.4, the process of calculating the update rule (44) also provides us the value of quantity $r_e(\frac{\pi}{2}M_n - \tau L)$ in each step. Therefore, the computational cost of verifying condition (47) is negligible. In order to set up our simple greedy algorithm, we quantify contribution of adding a new edge $e_i \in \mathcal{E}_c \backslash \mathcal{E}_s$ to the performance of the network by

$$\begin{aligned} h_{e_i}(\mathcal{E}_s) := \tilde{\rho}_{ss}\left(L + \sum_{e\in\mathcal{E}_s}\varpi(e)b_eb_e^{\mathrm{T}};\tau\right) \\ -\ \tilde{\rho}_{ss}\left(L + \sum_{e\in\mathcal{E}_s\cup\{e_i\}}\varpi(e)b_eb_e^{\mathrm{T}};\tau\right). \end{aligned} \tag{48}$$

In each step of the algorithm, the edge with maximum contribution to the performance is chosen and added to the coupling graph of the network. All steps of our method are summarized in Algorithm 1. According to Theorem 5.6, the augmented time-delay linear consensus network from Algorithm 1 is stable and has at most k new links.

Remark 5.7: Except some special cases where value of $h_e(\mathcal{E}_s)$ is identical for couple of links, where we should pick them randomly, the rest of the algorithm is deterministic.

Theorem 5.8: For linear consensus network (27) with a given candidate link $e \in \mathcal{E}_c \setminus \mathcal{E}_s$, the performance improvement by adding e to the network regardless of weight of e is upper bounded by

$$h_e(\mathcal{E}_s) \le \frac{r_e(L_sL_o^\dagger L_s)}{2r_e(L_s)} - \frac{c_1\tau}{r_e\left(\frac{\pi}{2}M_n - \tau L_s\right)}$$

Algorithm 1: Network Growing via Simple Greedy.

Initialize:
  $\mathcal{E}_s = \varnothing$
  $L_F = 0$
for $i = 1$ to k do:
  $e_i = \underset{\substack{e \in \mathcal{E}_c \setminus \mathcal{E}_s \\ \varpi(e) < w_s(L + L_F; e)}}{\arg\max} \; h_e(\mathcal{E}_s)$
  if $h_{e_i}(\mathcal{E}_s) \leq 0$:
    break
  $\mathcal{E}_s \leftarrow \mathcal{E}_s \cup \{e_i\}$
  $L_F = L_F + \varpi(e_i) b_{e_i} b_{e_i}^{\mathrm{T}}$
return $\mathcal{E}_s$

Algorithm 2: Network Growing via Random Greedy.

Initialize:
  $\mathcal{E}_s = \varnothing$
  $L_F = 0$
  add $2k-1$ dummy members* to $\mathcal{E}_c$
for $i = 1$ to k do:
  $M_i = \underset{\substack{M_i \subset \mathcal{E}_c \setminus \mathcal{E}_s, |M_i| = k \\ \varpi(e) < w_s(L + L_F; e), \quad \forall e \in M_i}}{\arg\max} \; \sum_{e \in M_i} h_e(\mathcal{E}_s)$
  repeat choose e_i randomly and uniformly from M_i
  $\mathcal{E}_s \leftarrow \mathcal{E}_s \cup \{e_i\}$
  $L_F = L_F + \varpi(e_i) b_{e_i} b_{e_i}^{\mathrm{T}}$
return $\mathcal{E}_s$

*These members are edges with zero weight.

where $c_1 = -0.01$ is a constant defined in Lemma 5.1 and $L_s = L + \sum_{e \in \mathcal{E}_s \cup \{e_i\}} \varpi(e) b_e b_e^T$.

Proof: From definition of h_e in (48) we have

$$h_e(\mathcal{E}_s) = \frac{r_e(L_s L_o^\dagger L_s)}{2\varpi(e)^{-1} + 2r_e(L_s)} - c_1 \tau^2 \varpi(e) + \frac{2\tau}{\pi} \frac{r_e\left((\frac{\pi}{2}M_n - \tau L_s) L_o^\dagger (\frac{\pi}{2}M_n - \tau L_s)\right)}{-(\varpi(e)\tau)^{-1} + r_e(\frac{\pi}{2}M_n - \tau L_s)}. \tag{49}$$

In addition, Theorem 5.6 states that $\varpi(e) < (\tau\, r_e(\frac{\pi}{2}M_n - \tau L_s))^{-1}$, and therefore, by multiplying both sides of the previous inequality by the nonnegative constant $-c_1\tau^2$, we obtain

$$-c_1 \tau^2 \varpi(e) \leq -c_1 \tau \left(r_e\left(\frac{\pi}{2}M_n - \tau L_s\right)\right)^{-1}. \tag{50}$$

Furthermore, stability of the network ensures $\frac{\pi}{2}M_n - \tau L_s \succeq 0$ from which we infer that $(\frac{\pi}{2}M_n - \tau L_s)^2 \succeq 0$ and consequently $r_e((\frac{\pi}{2}M_n - \tau L_s)^2) > 0$ for every $e \in \mathcal{E}_c$. Moreover, using Theorem 5.6, we can argue that

$$\frac{2\tau}{\pi} \frac{r_e\left((\frac{\pi}{2}M_n - \tau L_s) L_o^\dagger (\frac{\pi}{2}M_n - \tau L_s)\right)}{-(\varpi(e)\tau)^{-1} + r_e(\frac{\pi}{2}M_n - \tau L_s)} \leq 0. \tag{51}$$

A proof follows from combining inequalities (50) and (51) with identity (49). ■

In the literature, other variants of greedy algorithms, such as random greedy, are used in submodular problems with cardinality constraints [39]. Even though our performance measure is not a supermodular set function, our simulations show that random greedy works well and in some cases slightly outperform the simple greedy. Furthermore, their consistent outcome can be interpreted as a positive sign that both algorithms work fine. Based on Theorem 5.6, the augmented time-delay linear consensus network from Algorithm 2 is stable and has at most k new links.

C. Time Complexity Analysis

We provide a time complexity analysis based on the fastest state-of-the-art algorithms in the literature. First, we need to find the pseudoinverse for L and $\left(\frac{\pi}{2}M_n - \tau L\right)$, which has the complexity of $\mathcal{O}(n^3)$, and then, we calculate $L^\dagger L_o L^\dagger$ and $(\frac{\pi}{2}M_n - \tau L)^\dagger L_o (\frac{\pi}{2}M_n - \tau L)^\dagger$, which needs $\mathcal{O}(n^3) + \mathcal{O}(n^3 + mn^2)$. For all other steps, using the Sherman–Morrison formula [38] for rank-one update, we can find the update for $r_e(L)$, $r_e(LL_o^\dagger L)$, $r_e\left(\frac{\pi}{2}M_n - \tau L\right)$, and $r_e\left((\frac{\pi}{2}M_n - \tau L) L_o^\dagger (\frac{\pi}{2}M_n - \tau L)\right)$ for all $e \in \mathcal{E}_c$ in $\mathcal{O}(n^2)$. Then, finding the contribution for each link takes constant time for each link. In conclusion, the first step needs $\mathcal{O}(n^3)$ and the rest of the steps take $\mathcal{O}(n^2)$, which is less time compared to the method provided in [8], which considers network design in the absence of time-delay and the essence of their algorithms are similar to ours. The algorithm in [8] despite using the Sherman–Morrison formula eventually needs $\mathcal{O}(n)$ to find the contribution of each link in each step and since we have $\mathcal{O}(n^2)$ candidate links, their algorithm has time complexity of $\mathcal{O}(n^3)$ in all steps. The random greedy algorithm needs $\mathcal{O}(|\mathcal{E}_c| \log_2 k)$ more arithmetic operations than the simple greedy in each step since we have to find the top k contributing links.

VI. Improving Performance by Adjusting Feedback Gains

The second possible way to improve the performance of network (3) is by adjusting link weights in the underlying graph of the network. This option is domain specific and depends on the underlying dynamics of the network and on practical relevance of the problem. In the absence of time-delay, the optimal reweighting problem can be equivalently cast as the effective resistance minimization problem [6].

As it can be inferred from the previous section, although the problem of reweighting the coupling weights is a convex optimization problem, using the approximate performance measure greatly speeds up the rate of finding the optimal solution. Let denote the total weight of the links in the initial network by W_{total}. Then, the following SDP finds the Laplacian matrix for the optimal network in terms of the approximate performance measure:

$$\begin{aligned}
&\text{minimize} \quad \mathrm{Tr}\left[L_o X_1 + \frac{4}{\pi} L_o X_2 + c_1 \tau L_o \hat{L}\right] \\
&\text{subject to:} \quad \hat{L} = \sum_{e \in \mathcal{E}} \hat{w}(e) b_e b_e^T \\
&\qquad \sum_{e \in \mathcal{E}} \hat{w}(e) \leq W_{\text{total}} \\
&\qquad \hat{w}(e) \geq 0 \quad \text{for all } e \in \mathcal{E} \\
&\qquad \begin{bmatrix} X_1 & I \\ I & \tau\hat{L} + \frac{1}{n}J \end{bmatrix} \succeq 0, \begin{bmatrix} X_2 & I \\ I & \frac{\pi}{2}I_n - \tau\hat{L} \end{bmatrix} \succeq 0
\end{aligned} \tag{52}$$

where the optimization variables are matrices $X_1, X_2 \in \mathbb{R}^{n\times n}$ and nonnegative continuous variable $\hat{w}(e)$ as coupling weights for all $e \in \mathcal{E}$.

Remark 6.1: For the consensus network (4), the resulting network from solving the network reweighing problem (52) always have the same or better performance (with respect to $\tilde{\rho}_{ss}$) than the original network.

It is also possible to improve the performance measure of the network (3) by reweighting the links while keeping ratio of weight of every two links unchanged. The following theorem elaborates more on this approach.

Theorem 6.2: Suppose that for the consensus network (4), rows of the output matrix C span the disagreement space. Our objective is to improve performance of the network by scaling weights of all links by a constant $\kappa \in (0, \frac{\pi}{2\lambda_n \tau})$. Then, there exists a unique $\kappa^* > 0$ such that for all $\kappa \in (0, \frac{\pi}{2\lambda_n \tau})$. Moreover, the optimal κ^* belongs to the interval $[\frac{z^*}{\lambda_n}, \frac{z^*}{\lambda_2}]$, where λ_2 and λ_n are the second smallest and the largest eigenvalues of L.

Proof: For a fixed underlying graph, let $h : (0, \frac{\pi}{2\lambda_n \tau}) \to \mathbb{R}$ be a function where $g(\kappa) = \rho_{ss}(\kappa L; \tau)$. From Theorem 3.5, it follows that $g(\kappa)$ is a strictly convex function of κ. In addition, since $g(\kappa)$ is not a monotonic function of κ and is continuous on its domain, it must have an attainable minimum. Furthermore, by strict convexity of $g(\kappa)$, a unique positive κ^* exists where $g(\kappa)$ attains its minimum. In addition, if we scale the weights by any $\kappa < \frac{z^*}{\lambda_n}$, all the new eigenvalues will be on the left-hand side of the dashed line in Fig. 1 and therefore $\kappa = \frac{z^*}{\lambda_n}$ is better than any $\kappa \in (0, \frac{z^*}{\lambda_n})$. Similarly, if we scale the weights by any $\kappa > \frac{z^*}{\lambda_2}$, all the new eigenvalues will be on the right-hand side of the dashed line in Fig. 1 and thus $\kappa > \frac{z^*}{\lambda_2}$ will not be better than $\kappa = \frac{z^*}{\lambda_2}$. Finally, by convexity of $g(\kappa)$, we conclude that $\kappa^* \in [\frac{z^*}{\lambda_n}, \frac{z^*}{\lambda_2}]$. ∎

Since in the scaling method, we are not using rank-one update of the Laplacian matrix and we need to compute the spectrum of the underlying graph only once, we may use ρ_{ss} for finding κ^*. Thus, this method is the only approach in this paper that needs spectrum of the underlying graph, which can be computed in $\mathcal{O}(n^3)$. Knowing the Laplacian eigenvalues, κ^* can be found by exploiting simple techniques, such as golden section search method.

VII. Improving Performance by Feedback Sparsification

Suppose that we are required to remove some of the existing interconnection links in linear consensus network (4). Sparsification can potentially happen in practice for several reasons, including when there is a budget constraint on communication cost or an enforced security and/or privacy protocol [41] among the agents that limit each agent to communicate with certain number of neighbors. We assume that some of the Laplacian eigenvalues of the underlying graph are on the right-hand side of the dashed line in Fig. 1, i.e., $\lambda_i > \frac{z^*}{\tau}$ for some $i \geq 2$, where under this condition, eliminating some of the existing links can improve performance of the network. Edge elimination as a method to improve convergence speed of time-delay consensus networks was previously presented in [42], where authors measure the convergence rate through simulations in time domain.

The challenges of this approach are twofold. First, dropping links may break connectivity of the network. By the min–max theorem [43], dropping links from the underlying graph of the network (3) does not increase Laplacian eigenvalues, and therefore, if largest eigenvalue of the Laplacian is less than $\frac{\pi}{2\tau}$, after dropping links it will remain less than $\frac{\pi}{2\tau}$. Therefore, a failure in connectivity increases the number of connected components of the underlying graph. As a result, in the disconnected network, we will have connected components whose dynamics are decoupled and eigenvalues of each component will remain less than $\frac{\pi}{2\tau}$ as they are a subset of eigenvalues of the whole network. Therefore, each connected component will converge to its own consensus point. This implies that a failure in connectivity will result in an unbounded performance measure. Second, the sparsification problem is inherently combinatorial as we have to find a subset of the current interconnection links in the network and remove them. In the following, we propose remedies to these challenges.

To tackle the connectivity problem, we must ensure that the link that is being removed is not one of the cut edges; in other words, removing that specific link would not increase the number of connected components of the underlying graph. There exist bridge finding algorithms in an undirected graph, such as Tarjan's bridge-finding algorithm, which runs in linear time [44]. However, since in each step of our greedy algorithm, we have the effective resistance between every two nodes, we can effectively use this existing information to ensure that a selected candidate edge is not a cut edge. In order to avoid removing cut edges, we must ensure that for a candidate edge $e = \{i, j\}$, the following condition holds [45] $\omega(e) \neq \frac{1}{r_e(L)}$ where r_e is the effective resistance between nodes i and j.

In order to tackle combinatorial difficulty of sparsification, we utilize the following tailored greedy algorithm to sparsify the coupling graph of a given consensus network. In the algorithm below, $\mathcal{E}_s$ is the set of the coupling links that are chosen to be removed. For an edge $e_i \in \mathcal{E}_c \setminus \mathcal{E}_s$, we denote contribution of removing that edge to performance of the network by the following quantity:

$$\hat{h}_{e_i}(\mathcal{E}_s) := \tilde{\rho}_{ss}\left(L - \sum_{e\in\mathcal{E}_s} w(e) b_e b_e^T; \tau\right) - \tilde{\rho}_{ss}\left(L - \sum_{e\in\mathcal{E}_s\cup\{e_i\}} w(e) b_e b_e^T; \tau\right)$$

and in each step of our greedy, we choose an edge that is not a cut edge and its removal has maximum contribution to the performance. Algorithm 3 summarizes all steps of our greedy method.

We use $\tilde{\rho}_{ss}$ in Algorithm 3 in order to take advantage of the Sherman–Morrison formula and avoid costly eigendecomposition. Therefore, the process of adding the first link requires computation of $r_e(L)$, $r_e(LL_o^\dagger L)$, $r_e\left(\frac{\pi}{2}M_n - \tau L\right)$, and $r_e\left((\frac{\pi}{2}M_n - \tau L)L_o^\dagger(\frac{\pi}{2}M_n - \tau L)\right)$, which can be accomplished with $\mathcal{O}(n^3 + mn^2)$ arithmetic operations. Knowing the effective resistance between all nodes, by utilizing the aforementioned theorem, it takes $\mathcal{O}(1)$ operations to ensure that an edge is not a cut edge. As a result, cut-edge verification takes $\mathcal{O}(n^2)$ operation. For every other step of Algorithm 3, our method needs $\mathcal{O}(n^2)$ operations to update effective resistance matrices and ensure connectivity by not eliminating a cut edge.

Algorithm 3: Network Sparsification via Simple Greedy.

Initialize:
  $\mathcal{E}_s = \varnothing$
  $L_{_F} = 0$
for $i = 1$ to k do:
  $\mathcal{E}_b = \{e \,|\, w(e) = r_e^{-1}(L - L_{_F})\}$
  $e_i = \underset{e\in\mathcal{E}\setminus(\mathcal{E}_s\cup\mathcal{E}_b)}{\arg\max}\ \hat{h}_e(\mathcal{E}_s)$
  if $\hat{h}_{e_i}(\mathcal{E}_s) \le 0$:
    break
  $\mathcal{E}_s \leftarrow \mathcal{E}_s \cup \{e_i\}$
  $L_{_F} = L_{_F} - w(e_i) b_{e_i} b_{e_i}^{\mathrm{T}}$
return $\mathcal{E}_s$

Remark 7.1: Our notion of sparsification is basically different from spectral sparsification of [46]. In our approach, we only eliminate some of the coupling links without reweighting the remaining ones. In addition, our goal is not to create a sparse network with similar performance, but to achieve better performance by reshaping the spectrum of the underlying graph of the network.

VIII. Sensitivity Analysis of the Feedback Structure

As we discussed earlier, there are three ways to improve performance. For a given network, our remaining task is to determine which one of the proposed methods should be employed to improve performance of the network. Our reweighing procedure in Section VI never deteriorates network's performance, i.e., either it improves it or does not change the weights.

Theorem 8.1: Suppose that e is the edge between node i and mode j in the coupling graph of network (4). Then, sensitivity of $\tilde{\rho}_{ss}$ with respect to weight of link e is equal to

$$\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} = \frac{c_1\tau^2}{2} r_e(L_o^\dagger) - \frac{1}{2} r_e(L L_o^\dagger L) + \frac{2\tau^2}{\pi} r_e\left(\left(\frac{\pi}{2}M_n - \tau L\right) L_o^\dagger \left(\frac{\pi}{2}M_n - \tau L\right)\right) \tag{53}$$

where $c_1 = -0.01$ is a constant defined in Lemma 5.1.

Proof: Taking partial derivative of $\tilde{\rho}_{ss}$ with respect to weight of edge e, we have

$$\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} = \frac{d}{dw(e)}\left(\frac{1}{2}\,\mathrm{Tr}\left[L_o L^\dagger + \frac{4\tau}{\pi} L_o \left(\frac{\pi}{2}M_n - \tau L\right)^\dagger + c_1\tau^2 L_o L\right]\right). \tag{54}$$

Then, we find derivative of all terms in the right-hand side of the above equation as follows:

$$\frac{d\,\mathrm{Tr}[L_o L^\dagger]}{dw(e)} = -\,\mathrm{Tr}[L_o L^\dagger(\chi_i - \chi_j)(\chi_i - \chi_j)^{\mathrm{T}} L^\dagger] = -r_e(L L_o^\dagger L),$$

$$\frac{d\,\mathrm{Tr}[L_o(\frac{\pi}{2}M_n - \tau L)^\dagger]}{dw(e)} = \tau\,\mathrm{Tr}\left[L_o\left(\frac{\pi}{2}M_n - \tau L\right)^\dagger(\chi_i - \chi_j)(\chi_i - \chi_j)^T(\frac{\pi}{2}M_n - \tau L)^\dagger\right] = \tau r_e\left(\left(\frac{\pi}{2}M_n - \tau L\right) L_o^\dagger \left(\frac{\pi}{2}M_n - \tau L\right)\right).$$

On substituting above identities in (54) we obtain

$$\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} = \frac{c_1\tau^2}{2} r_e(L_o^\dagger) - \frac{1}{2} r_e(L L_o^\dagger L) + \frac{2\tau^2}{\pi} r_e\left(\left(\frac{\pi}{2}M_n - \tau L\right) L_o^\dagger \left(\frac{\pi}{2}M_n - \tau L\right)\right).$$

Equality (53) turns out to be useful in distinguishing whether sparsification or growing the underlying graph can be effective methods to improve the performance measure. Although it is also possible to figure out whether removing or adding interconnection can improve the performance by finding the spectrum of the Laplacian matrix in each step, (53) is important since it does not need eigendecomposition and can be updated in each iteration in $\mathcal{O}(n^2)$. Therefore, if adding and removing connections were an option, adding connections will improve the performance only if $\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} < 0$ for some edge $e \in \mathcal{E}_c$. Similarly, sparsifying the network can be useful only if $\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} > 0$ for some edge $e \in \mathcal{E}$. Besides, we can use this approach as a heuristic for the rewiring problem.

When we are given a set of weightless candidate links, the contribution of each link to the performance cannot be evaluated. If size of the problem (i.e., the number of nodes and size of candidate set) is small and we are not concerned about sparsity of the solution, we can use SDP given by (35)–(39) to find optimal link weights. However, if sparsity is an issue and our objective is to add at most k new links, we can use identity (53) for adding new links. For the first link, the procedure includes finding $e_1 = \arg\min_{e\in\mathcal{E}_c} \frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)}$; this is the link for which the performance measure has the most sensitivity with respect to its weight. As it was discussed earlier, we must have $\frac{d\tilde{\rho}_{ss}(L;\tau)}{dw(e)} < 0$. Otherwise, the procedure will be terminated as adding new edges cannot improve $\tilde{\rho}_{ss}$. In addition, we can find the best weight for the selected edge e_1 by minimizing (45) over $\varpi(e_1)$ subject to $0 \le \varpi(e_1) < w_s(L; e_1)$, which can be done in constant time. Then, we initialize $\mathcal{E}_s = \{e_1\}$. In order to identify the ith link, we set

$$e_i = \underset{e\in\mathcal{E}_c\setminus\mathcal{E}_s}{\arg\min} \frac{d\tilde{\rho}_{ss}\left(L + \sum_{e\in\mathcal{E}_s}\varpi(e) b_e b_e^{\mathrm{T}};\tau\right)}{dw(e)}$$

and maximize $h_{e_i}(\mathcal{E}_s)$ over $\varpi(e_i)$. Subsequently, we add e_i to $\mathcal{E}_s$ and procedure continues until we have added k edges or $\frac{d\tilde{\rho}_{ss}\left(L + \sum_{e\in\mathcal{E}_s}\varpi(e) b_e b_e^{\mathrm{T}};\tau\right)}{dw(e)} \ge 0$ for all $e \in \mathcal{E}_c \setminus \mathcal{E}_s$.

IX. Numerical Examples

In this section, we consider the following numerical examples to demonstrate utility and veracity of our theoretical results, where the data for graph Laplacians of all examples can be downloaded from the following link:

http://goo.gl/PvwPKC

Example 9.1: As a means to compare performance of a network in the presence and absence of time-delay, we show that in the presence of delay, adding a link in two different locations in the network has contrasting effect on performance of the network. Unweighted graphs in Fig. 2(b) and Fig. 2(c) are constructed by adding one link to distinct locations of the graph shown in Fig. 2(a). When $\tau = 0.235$, performance measure of

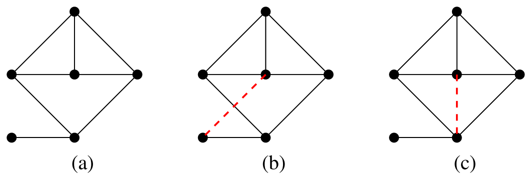

Fig. 2. (a) Graph 1, (b) Graph 2, and (c) Graph 3 are unweighted graphs. (b) and (c) are constructed by adding a link to (a).

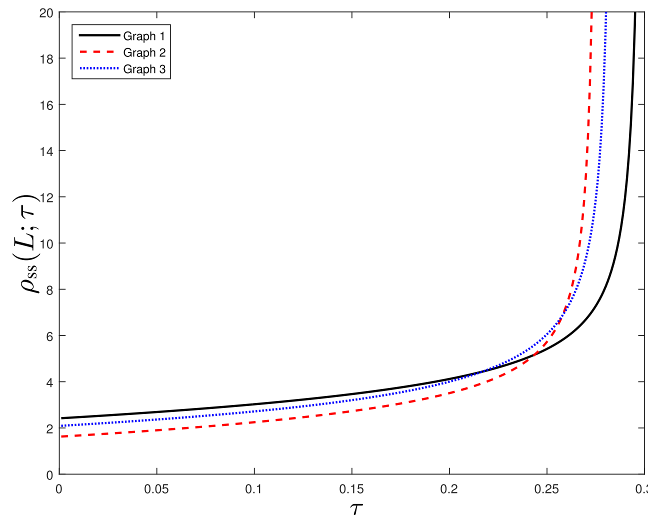

Fig. 3. Comparing performance of three different topologies shown in Fig. 2 as a function of time-delay.

the network with underlying graph of Fig. 2(b) is better than the original network and both are better than the network with underlying graph of Fig. 2(c). Nevertheless, without the delay, networks with underlying graph Fig. 2(b) and Fig. 2(c) perform better than the original network in terms of noise propagation quantified by $\mathcal{H}_2$-norm. In order to further clarify effect of connectivity in the presence or absence of time-delay, in Fig. 3, we drew performance of the three aforementioned networks as a function of time-delay. It is noteworthy that when $\tau < 0.2$, consensus network with the coupling graph given in Fig. 2(d) has a better performance than a network with the coupling graph given in Fig. 2(a). Whereas, as the time-delay increases, the network with underlying graph in Fig. 2(a) starts to outperform the network with graph given in Fig. 2(c).

Example 9.2: Consider the arbitrary network (3) with 125 nodes and initially 250 unweighted links are given by Fig. 4 in the presence of $\tau = 0.017$ delay. We design an optimal topology for the network using SDP relaxation, simple greedy, and random greedy. Then, we compare them with the hard limit to check how close each method can get to the theoretical lower bound of the solution. We add 7500 new links using the SDP method and 2439 new links using the simple greedy algorithm. We see that network's square of $\mathcal{H}_2$-norm performance is improved by 88.8 %, from 29.28 to 3.27 using simple greedy. From the result of Theorem 4.1, the value of the hard limit for the performance of the network is 3.237 and we know that the global optimal for the problem is greater than the hard limit. It should be further noted that there exists only 1.2% difference between hard limit and the new performance of the network generated by our simple greedy algorithm and even smaller gap for SDP. We observe that the result of random greedy can be different

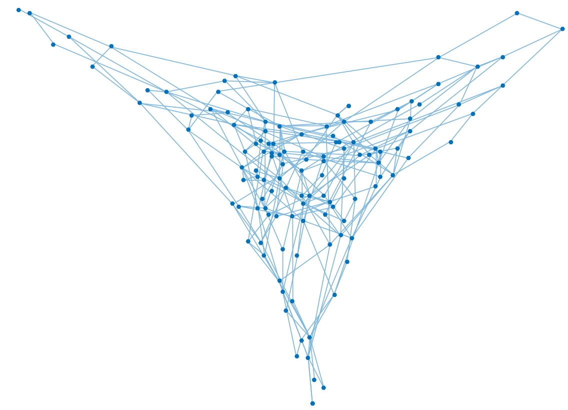

Fig. 4. Arbitrary unweighted graph with 125 nodes and 250 edges.

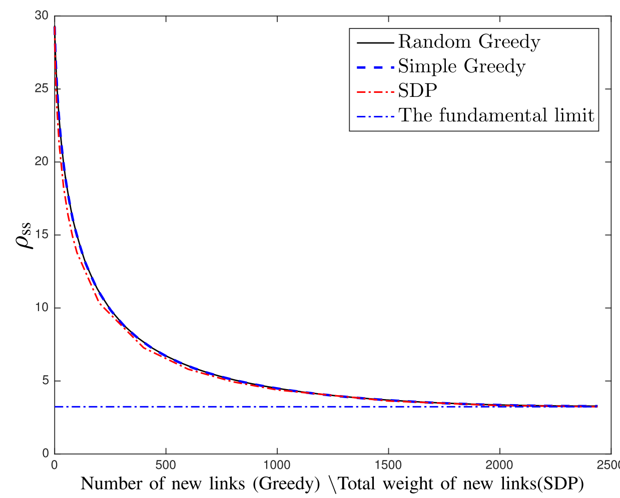

Fig. 5. Improving the performance of the network given in Fig. 4 by adding new interconnections through SDP and greedy algorithms.

in each run of the algorithm. It is noteworthy that in the case of this example, although the final networks generated by random greedy and simple greedy are very different, eventually the difference between performance of the network generated by them is not very different, as it can be shown in Fig. 5. Our simulations confirm that our proposed simple greedy performs near optimal for generic time-delay linear consensus networks. In Example 9.5, we construct a specific network that by which it is argued that random greedy may outperform simple greedy by a considerable margin.

Example 9.3: Here, we want to evaluate the strength of establishing new interconnections using our greedy algorithm. To that end, we use our greedy algorithm to add edges to a randomly generated graph given in Fig. 8 that has 10 nodes and 15 edges initially. The time-delay is set equal to $\tau = 0.05$. Moreover, we suppose that set of candidate edges is complement of the set of initial edges. We intend to establish up to 16 new interconnections. As it is depicted in Fig. 9, the algorithm yields extremely good results. Our simulations results assert that our suggested simple greedy provide near-optimal solution to time-delay linear consensus networks with generic graph topologies.

Example 9.4: Let us consider linear consensus network (3) with 800 nodes and initially 2×10^4 unweighted links given by Fig. 6 in the presence of $\tau = 0.019$ delay. Our goal is to

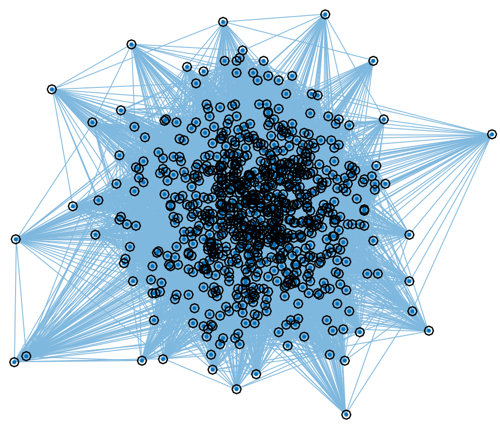

Fig. 6. Arbitrary unweighted graph with 800 nodes and 20000 edges.

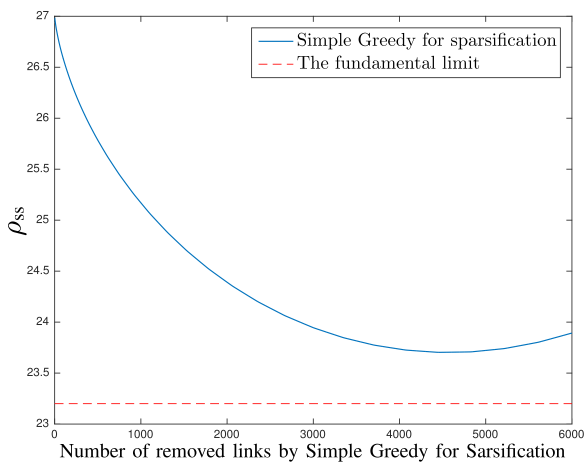

Fig. 7. Improving performance of the network given in Fig. 6 by removing interconnections through sparsification.

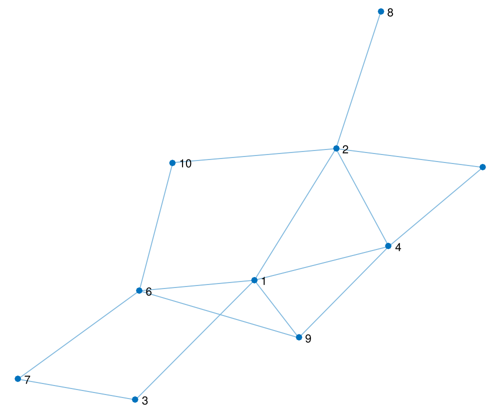

Fig. 8. Arbitrary unweighted graph with 10 nodes and 15 edges.

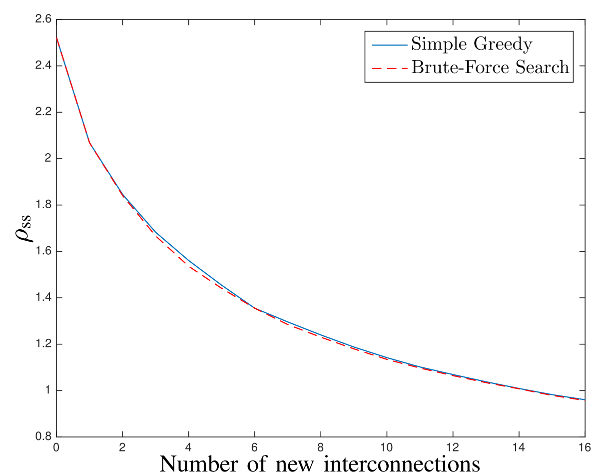

Fig. 9. Improving performance of the network given in Fig. 8 by adding new interconnections.

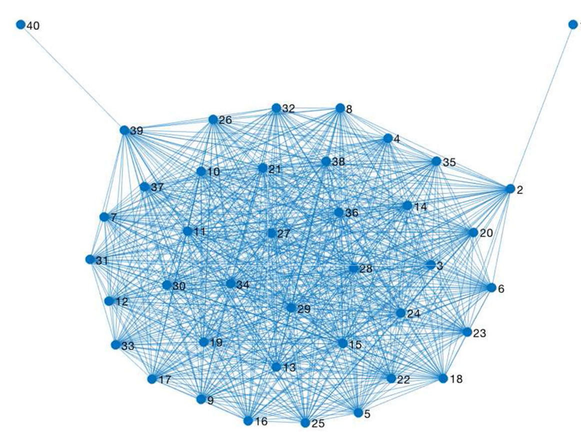

Fig. 10. Arbitrary graph with 40 nodes with uniform weight 1. Every subset of nodes $\{2, \ldots, 39\}$ forms a clique, i.e., the subgraph induced by them is a complete graph.

remove links from the coupling graph of the network using our sparsification Algorithm 3 in order to improve the performance. We compare the best achieved performance with the hard limit, to see how close we can get to the lower bound of the solution. We removed 6×10^3 links by executing Algorithm 3. In Fig. 7, we observe that the best performance is achieved by removing 4461 links and the network's square of $\mathcal{H}_2$-norm performance is improved by 12%, reaching 23.7, which was initially 27. Using inequality (18), the hard limit for the performance of the network is 23.2 and we know that the optimal solution for the problem is greater than or equal to the hard limit. It is noteworthy that there exists only 2.2% difference between hard limit and the best achieved performance.

Example 9.5: Suppose that the linear consensus network (3) with underlying graph in Fig. 10 and time-delay $\tau = 1.852 \times 10^{-2}$ is given. Our aim is to show that growing the network by random greedy sometimes outperforms the simple greedy algorithm. The set of candidate links is $\mathcal{E}_c = \mathcal{E}_{\mathcal{K}_n} \setminus \mathcal{E}$, where $\mathcal{E}_{\mathcal{K}_n}$ is the set of all edges in complete graph on n nodes $\mathcal{K}_n$. Let us set $\varpi(e_1) = 40$ for $e_1 = \{1, 40\}$ and $\varpi(e) = 1$ for all other edges $e \in \mathcal{E}_c \setminus \{e_1\}$. The performance measure of the initial network is 2.0686. The value of the performance measure for the resulting network from Algorithm 1 is reduced to 1.5195, whereas by applying Algorithm 2, the performance measure can be reduced up to 1.1038 with high probability. In this specific example, we observe that the simple greedy improves the performance 47% less than the random greedy. We would like to mention that if we set $\varpi(e_1) = 1$, the simple greedy provides us with the optimal solution.

X. Discussion and Conclusion

We studied $\mathcal{H}_2$-norm performance of noisy first-order time-delay linear consensus networks from a spectral graph theoretical point of view. It is shown that this measure is convex with respect to weights and increasing with respect to time-delay. We propose low-complexity methods to improve the performance of such networks, where the fastest of which has cubic time algorithm capable of generating sparse solutions. The design problems discussed in this paper can also be formulated as SDP problems, where solving such convex problem requires costly $\mathcal{O}(n^3 + |\mathcal{E}|^3)$ operations for each iteration. This is why we have favored greedy algorithms to design time-delay linear consensus networks that offer significantly lower time complexities.

The focus of this paper has been on time-delay linear consensus networks with state-space matrices $(-L, I, C, 0)$, where C is an arbitrary output matrix that are orthogonal to the vector of all ones. Our methodology can also handle time-delay networks with state-space matrices $(-L, B, M_n, 0)$. Using duality of controllability and observability, one can show that $\mathcal{H}_2$-norm of the following three networks with state-space matrices $(-L, B, M_n, 0)$, $(-L, M_n, B^{\mathrm{T}}, 0)$, and $(-L, I, B^{\mathrm{T}}, 0)$ are equal and they are bounded if vector of all ones is in the left nullspace of matrix B and the time-delay is less than the time-delay margin.

In this paper, we assumed that time-delay is uniform across the network. Performance of consensus networks with nonuniform delay is a possible direction to generalize our results.

References

[1] M. Siami and N. Motee, "Fundamental limits and tradeoffs on disturbance propagation in linear dynamical networks," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4055–4062, Dec. 2016.

[2] B. Bamieh, M. R. Jovanović, P. Mitra, and S. Patterson, "Coherence in large-scale networks: Dimension-dependent limitations of local feedback," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2235–2249, Sep. 2012.

[3] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *Proc. Amer. Control Conf.*, 2010, pp. 6312–6317.

[4] D. Zelazo and M. Mesbahi, "Edge agreement: Graph-theoretic performance bounds and passivity analysis," *IEEE Trans. Autom. Control*, vol. 56, no. 3, pp. 544–555, Mar. 2011.

[5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[6] A. Ghosh, S. Boyd, and A. Saberi, "Minimizing effective resistance of a graph," *SIAM Rev.*, vol. 50, no. 1, pp. 37–66, 2008.

[7] M. Siami and N. Motee, "Tractable approximation algorithms for the NP-hard problem of establishing new interconnections in linear consensus networks," in *Proc. Amer. Control Conf.*, 2016, pp. 6429–6434.

[8] T. Summers, I. Shames, J. Lygeros, and F. Dorfler, "Topology design for optimal network coherence," in *Proc. Eur. Control Conf.*, 2015, pp. 575–580.

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

[10] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is NP-hard," *Oper. Res. Lett.*, vol. 36, no. 6, pp. 677–679, 2008.

[11] U. Münz, A. Papachristodoulou, and F. Allgöwer, "Delay robustness in consensus problems," *Automatica*, vol. 46, no. 8, pp. 1252–1265, 2010.

[12] I. Lestas and G. Vinnicombe, "Scalable decentralized robust stability certificates for networks of interconnected heterogeneous dynamical systems," *IEEE Trans. Autom. Control*, vol. 51, no. 10, pp. 1613–1625, Oct. 2006.

[13] U. Jönsson, C.-Y. Kao, and H. Fujioka, "Low dimensional stability criteria for large-scale interconnected systems," in *Proc. Eur. Control Conf.*, 2007, pp. 2741–2747.

[14] U. T. Jonsson and C.-Y. Kao, "A scalable robust stability criterion for systems with heterogeneous LTI components," *IEEE Trans. Autom. Control*, vol. 55, no. 10, pp. 2219–2234, Oct. 2010.

[15] A. Papachristodoulou and A. Jadbabaie, "Synchronization in oscillator networks: Switching topologies and non-homogeneous delays," in *Proc. 44th IEEE Conf. Decis. Control*, 2005, pp. 5692–5697.

[16] S. Z. Khong, E. Lovisari, and A. Rantzer, "A unifying framework for robust synchronization of heterogeneous networks via integral quadratic constraints," *IEEE Trans. Autom. Control*, vol. 61, no. 5, pp. 1297–1309, May 2016.

[17] C. Somarakis and J. S. Baras, "Delay-independent stability of consensus networks with application to flocking," *IFAC-PapersOnLine*, vol. 48, no. 12, pp. 159–164, 2015.

[18] W. Qiao and R. Sipahi, "A linear time-invariant consensus dynamics with homogeneous delays: Analytical study and synthesis of rightmost eigenvalues," *SIAM J. Control Optim.*, vol. 51, no. 5, pp. 3971–3992, 2013.

[19] M. Rafiee and A. M. Bayen, "Optimal network topology design in multi-agent systems for efficient average consensus," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 3877–3883.

[20] D. Hunt, "Network synchronization in a noisy environment with time delays," Ph.D. dissertation, Grad. Fac. Rensselaer Polytech. Inst., Troy, NY, USA, 2012.

[21] S. Hod, "Analytic treatment of the network synchronization problem with time delays," *Phys. Rev. Lett.*, vol. 5, 2010, Art. no. 208701.

[22] D. Hunt, G. Korniss, and B. K. Szymanski, "Network synchronization in a noisy environment with time delays: Fundamental limits and trade-offs," *Phys. Rev. Lett.*, vol. 105, no. 6, 2010, Art. no. 068701.

[23] S. Dezfulian, Y. Ghaedsharaf, and N. Motee, "On performance of time-delay linear consensus networks with directed interconnection topologies," in *Proc. Amer. Control Conf.*, 2018, pp. 4177–4182.

[24] H. Moradian and S. S. Kia, "Dynamic average consensus in the presence of communication delay over directed graph topologies," in *Proc. Amer. Control Conf.*, 2017, pp. 4663–4668.

[25] Y. Ghaedsharaf, M. Siami, C. Somarakis, and N. Motee, "Interplay between performance and communication delay in noisy linear consensus networks," in *Proc. Eur. Control Conf.*, 2016, pp. 1703–1708.

[26] Y. Ghaedsharaf and N. Motee, "Complexities and performance limitations in growing time-delay noisy linear consensus networks," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 228–233, 2016.

[27] Y. Ghaedsharaf and N. Motee, "Performance improvement in time-delay linear consensus networks," in *Proc. Amer. Control Conf.*, 2017, pp. 2345–2350.

[28] M. Siami, S. Bolouki, B. Bamieh, and N. Motee, "Centrality measures in linear consensus networks with structured network uncertainties," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 924–934, Sep. 2018.

[29] Y. Ghaedsharaf, M. Siami, C. Somarakis, and N. Motee, "Eminence in presence of time-delay and structured uncertainties in linear consensus networks," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 3218–3223.

[30] J. Doyle, K. Glover, P. Khargonekar, and B. Francis, "State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems," *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.

[31] Y. Ghaedsharaf, M. Siami, C. Somarakis, and N. Motee, "Performance Improvement in Noisy Linear Consensus Networks with Time-Delay," Extended version. [Online]. Available: https://arxiv.org/pdf/1810.08287.pdf

[32] D. Hunt, B. Szymanski, and G. Korniss, "Network coordination and synchronization in a noisy environment with time delays," *Phys. Rev. E*, vol. 86, no. 5, 2012, Art. no. 056114.

[33] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 2692–2697.

[34] C. Davis, "All convex invariant functions of hermitian matrices," *Archiv Der Mathematik*, vol. 8, no. 4, pp. 276–278, 1957.

[35] M. Siami and N. Motee, "Schur-convex robustness measures in dynamical networks," in *Proc. Amer. Control Conf.*, 2014, pp. 5198–5203.

[36] D. Hunt, G. Korniss, and B. Szymanski, "The impact of competing time delays in coupled stochastic systems," *Phys. Lett. A*, vol. 375, no. 5, pp. 880–885, 2011.

[37] M. S. Pranić and L. Reichel, "Recurrence relations for orthogonal rational functions," *Numerische Mathematik*, vol. 123, no. 4, pp. 629–642, 2013.

[38] W. W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, no. 2, pp. 221–239, 1989.

[39] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, "Submodular maximization with cardinality constraints," *Proc. 25th annu. ACM-SIAM symp. Discrete algorithms*, 2014, pp. 1433–1452.

[40] N. Rezazadeh and S. S. Kia, "Privacy preservation in a continuous-time static average consensus algorithm over directed graphs," in *Proc. Amer. Control Conf.*, 2018.

[41] M. H. Koh and R. Sipahi, "Achieving fast consensus by edge elimination in a class of consensus dynamics with large delays," in *Proc. Amer. Control Conf.*, 2016, pp. 5364–5369.

[42] G. Teschl, *Mathematical Methods in Quantum Mechanics*, vol. 157. Providence, RI, USA: Amer. Math. Soc., 2014.

[43] R. E. Tarjan, "A note on finding the bridges of a graph," *Inf. Process. Lett.*, vol. 2, no. 6, pp. 160–161, 1974.

[44] D. J. Klein and M. Randić, "Resistance distance," *J. Math. Chem.*, vol. 12, no. 1, pp. 81–95, 1993.

[45] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1913–1926, 2011.

[46] E. Jarlebring, J. Vanbiervliet, and W. Michiels, "Characterizing and computing the $\mathcal{H}_2$ norm of time-delay systems by solving the delay Lyapunov equation," *IEEE Trans. Autom. Control*, vol. 56, no. 4, pp. 814–825, Apr. 2011.

[47] V. L. Kharitonov and E. Plischke, "Lyapunov matrices for time-delay systems," *Syst. Control Lett.*, vol. 55, no. 9, pp. 697–706, 2006.

[48] E. Plischke, "Transient effects of linear dynamical systems," Ph.D. dissertation, Math. Inform. Universität Bremen, Bremen, Germany, 2005.

[49] D. Varberg, *Convex Functions Pure and Applied Mathematics: A Series of Monographs and Textbooks*. New York, NY, USA: Elsevier, 1973.

[50] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge University Press, 2012.

[51] X.-D. Zhang and R. Luo, "The spectral radius of triangle-free graphs," *Australas. J. Combinatorics*, vol. 26, pp. 33–40, 2002.

Yaser Ghaedsharaf (S'16) received the B.Sc. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 2013. He is currently working toward the Ph.D. degree at the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA.

His research interests include machine learning, analysis and optimal design of networked control systems with applications in distributed control and cyber-physical systems, and robotics.

Mr. Ghaedsharaf was the Runner Up for the Best Student Paper Award in the 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems in 2016. Furthermore, he was the recipient of the Rossin College of Engineering Doctoral Fellowship in 2016, the RCEAS fellowship award in 2018, and the Mountaintop Research Fellowship in 2018.

Milad Siami (S'12–M'18) received the dual B.Sc. degrees in electrical engineering and pure mathematics and the M.Sc. degree in electrical engineering, both from Sharif University of Technology, Tehran, Iran, in 2009 and 2011, respectively, and the M.Sc. and Ph.D. degrees in mechanical engineering from Lehigh University, Bethlehem, PA, USA, in 2014 and 2017, respectively.

From 2009 to 2010, he was a research student with the Department of Mechanical and Environmental Informatics, Tokyo Institute of Technology, Tokyo, Japan. He is currently a Postdoctoral Associate with the Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include distributed control systems, distributed optimization, and applications of fractional calculus in engineering.

Dr. Siami was the recipient of the Gold Medal of National Mathematics Olympiad, Iran (2003) and the Best Student Paper Award at the 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems (2015). Moreover, he was awarded the RCEAS Fellowship (2012), Byllesby Fellowship (2013), Rossin College Doctoral Fellowship (2015), and Graduate Student Merit Award (2016) at Lehigh University.

Christoforos Somarakis (S'09–M'15) received the B.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 2007, and the M.S. and Ph.D. degrees in applied mathematics from the University of Maryland at College Park, College Park, MD, USA, in 2012 and 2015, respectively.

He is currently a Research Scientist with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA. His research interests include analysis and optimal design of networked control systems with applications in distributed control and cyber-physical systems.

Nader Motee (S'99–M'08–SM'13) received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2000, and the M.Sc. and Ph.D. degrees in electrical and systems engineering from the University of Pennsylvania, Philadelphia, PA, USA, in 2006 and 2007, respectively.

From 2008 to 2011, he was a Postdoctoral Scholar with the Control and Dynamical Systems Department, Caltech, Pasadena, CA, USA. He is currently an Associate Professor with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, USA. His current research area is distributed dynamical and control systems with particular focus on issues related to sparsity, performance, and robustness.

Dr. Motee is a past recipient of several awards including the 2008 AACC Hugo Schuck best paper award, the 2007 ACC best student paper award, the 2008 Joseph and Rosaline Wolf best thesis award, the 2013 Air Force Office of Scientific Research Young Investigator Program award, the 2015 NSF Faculty Early Career Development (CAREER) award, and the 2016 Office of Naval Research Young Investigator Program award.