Performance Analysis and Optimal Design of Time-Delay Directed Consensus Networks

Shima Dezfulian, Yaser Ghaedsharaf and Nader Motee

***Abstract*—The $\mathcal{H}_2$ performance of a class of noisy time-delay consensus networks with directed interconnection topologies is considered. For networks with normal Laplacian, we derive a closed-form expression for the performance measure and study its functional properties. The scaling properties of the $\mathcal{H}_2$-norm is investigated for some families of networks. Finally, the problem of tuning feedback gains of a network with fixed topology in order to improve its performance is considered. Our proposed design method can handle networks with several thousands state variables.**

I. Introduction

All real-world control systems experience time-delay and noise when closing the loop using sensory data. The presence of these undesired factors usually make analysis and design procedures more challenging [1]. Intractability of analysis and design problems as well as emergence of fundamental limits and tradeoffs, as side effects of time-delay, become more evident in the context of networked systems. It is shown that $\mathcal{H}_2$ performance of linear consensus networks deteriorate by increasing time-delay and is a non-monotone function of design parameters [2]. Intrinsic tradeoffs emerge between magnitude of extreme fluctuations, time-delay, statistics of noise, and network connectivity [3], [4], which reveals that time-delay is indeed origin of network fragility to certain systemic events. The goal of this paper is to measure quality of achieving consensus in presence of time-delay and noise for networks with directed interconnection topologies, and then apply our results to design optimal network topologies.

Stability analysis of time-delay linear time-invariant (LTI) systems is presented in [5], [6]. In [7] and [8], the authors characterize necessary and sufficient conditions for achieving consensus in time-delay consensus network with undirected graph topologies. Some sufficient conditions for average-consensus of undirected networks, with possibly constant, time-varying, uniform and non-uniform time-delay, are investigated in [9]. When there is no time-delay, [10] and [11] provide necessary and sufficient conditions for achieving consensus in directed networks.

In [12], delay-Lyapunov equations are used to quantify $\mathcal{H}_2$-norm of general time-delay LTI systems, where an explicit expression for the $\mathcal{H}_2$-norm is obtained for systems with commensurate delays, i.e., time-delays that are all integer multiples of a basic delay. The $\mathcal{H}_2$-norm of a delay-free directed consensus network is used as a robustness measure in [13], where a closed-form expression for $\mathcal{H}_2$-norm is obtained for networks with normal Laplacian matrices. The authors of [14] and [15] evaluate $\mathcal{H}_2$ performance of delay-free first- and second-order consensus networks with undirected topologies. In [16], it is shown that $\mathcal{H}_2$-norm of a time-delay undirected consensus network can be characterized explicitly in terms of time-delay and Laplacian eigenvalues and approximated effciently by another spectral function that can be used to enhance the network performance via growing, sparsification, and reweighing procedures. Obtaining closed-form expressions for performance measures in time-delay networks opens up new possibilities to reduce computational complexity of design algorithms [2], [4].

In our first contribution in Section V, we derive a closed-form expression for $\mathcal{H}_2$-norm of time-delay consensus networks with directed coupling topologies and normal Laplacians. It is shown that the $\mathcal{H}_2$ performance measure is an increasing function of the time-delay and it has a non-monotonic behavior with respect to feedback gains. When there is no time-delay, it is known that the $\mathcal{H}_2$ performance of consensus networks (with undirected graph) can be improved indefinitely by increasing the value of feedback gains. This property does not hold in presence of time-delay. We prove that allowing all-to-all communication between agents to exchange their state information with certain feedback gains, that depends on time-delay, will result in the best achievable $\mathcal{H}_2$ performance. Then, we study scaling properties of the performance measure (as a function of network size) for some families of networks.

In our second contribution in Section VI, we consider the problem of improving performance of a time-delay directed network with fixed topology through adjusting feedback gains. The design problem can be cast as an optimization problem that can handle both consensus and average-consensus networks. We exploit structural properties of the involving system matrices in order to reduce computational complexity of the design problem. To compute the performance measure (i.e., the cost function) and its gradient, we utilize iterative methods to solve the corresponding large-scale time-delay Lyapunov equations [17]. The common approach is to apply Krylov subspace method along with a pre-conditioner to solve the resulting time-delay Lyapunov equations [18], where one can compute $\mathcal{H}_2$-norm of a time-delay LTI system with around one thousand state variables. The pre-conditioner proposed by [18] is not applicable to our network design problem. Therefore, we propose a new pre-conditioner that performs well when the ratio of time-delay to time-delay margin is sufficiently small. Our proposed method enables us to compute and optimize performance of time-delay consensus networks with up to several thousands states variables.

This paper is an evolved version of [19], that presents several new results, including materials of Section VI and

S. Dezfulian, Y. Ghaedsharaf and N. Motee are with the Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA, 18015 USA

simulations, and provides a comprehensive discussion of the subject.

II. Preliminaries and Definitions

$\mathbb{R}^+(\mathbb{R}^{++})$ is the set of non-negative (positive) real numbers and $\mathbb{C}$ is the set of complex numbers. A complex number c is denoted by $c = \Re\{c\} + \Im\{c\}\jmath$, where $\Re\{c\}$ and $\Im\{c\}$ denote the real and imaginary parts of c, respectively. Also, $\jmath$ is the imaginary unit, so $\jmath^2 = -1$. $\mathcal{I}$ is a shorthand notation for the set $\{1, 2, \ldots, n\}$. Vector $\mathbf{e}_i \in \mathbb{R}^n$ is a unit vector which its i^{th} entry is equal to one and the rest of its entries are equal to zero. $\mathbf{1}_n$ is the vector of all ones. For a matrix M, transpose, conjugate, conjugate transpose, and Moore-Penrose pseudoinverse are shown by M^T, $\bar{M}$, M^{H}, and $M^{\dagger}$, respectively. $[M]_{ij}$ indicates the ij^{th} entry of matrix M. I_n is the identity matrix in $\mathbb{R}^{n\times n}$, and $0_{n\times n}$ is the matrix of all zeros. Centering matrix is defined as $M_n = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$. Kronecker product of matrix M and matrix N is denoted by $M \otimes N$. Vectorization of matrix M is shown by $\mathrm{vec}(M)$, which is a column vector. Trace of matrix M is denoted by $\mathrm{Tr}(M)$. Matrix $M \in \mathbb{C}^{n\times n}$ is unitary if $MM^{\mathrm{H}} = M^{\mathrm{H}}M = I_n$. Matrix $M \in \mathbb{C}^{n\times n}$ is normal if $MM^{\mathrm{H}} = M^{\mathrm{H}}M$.

Big O notation is shown by $\mathcal{O}$, e.g., $f(x) = \mathcal{O}\big(g(x)\big)$ means that there exists some positive real number κ and a real number x_0 such that for every x greater than or equal to x_0, $|f(x)| < \kappa|g(x)|$. A weighed directed graph $\mathcal{G}$ is defined by a tuple $(\mathbf{V}, \mathbf{E}, w)$. $\mathbf{V}$ is the set of nodes, $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ is the set of edges and $w : \mathbf{V} \times \mathbf{V} \to \mathbb{R}^+$ is a weight function such that

$$w\big((i,j)\big) = \begin{cases} w_{ij} & \text{if } (i,j) \in \mathbf{E} \\ 0 & \text{otherwise} \end{cases},$$

where w_{ij} is the weight of edge $(i,j) \in \mathbf{E}$. Let k_{ij} to be the shorthand notation for $w\big((i,j)\big)$ for all $(i,j) \in \mathbf{V} \times \mathbf{V}$. For any edge $(i,j) \in \mathbf{E}$, node j is its head and node i is its tail. We assume that nodes do not have self-loops and there are no parallel edges from one node to another node. A directed path in graph $\mathcal{G}$ is a sequence of distinct nodes of the graph which are connected to each other by a sequence of edges [20]. A graph $\mathcal{G}$ has a globally reachable node if there exists a node such that there is a directed path from any other node of the graph $\mathcal{G}$ to that node [21]. A node j is in the neighborhood of node i, if $(i,j) \in \mathbf{E}$; therefore, set of neighbors of node i is defined as $\mathbf{N}_i = \{j \in \mathbf{V} | (i,j) \in \mathbf{E}\}$. In-degree of the node i is $d_i^{in} = \sum_{j=1}^{n} k_{ji}$, likewise, out-degree of node i is $d_i^{out} = \sum_{j=1}^{n} k_{ij}$. A graph is balanced if and only if all of its nodes have the same in-degree and out-degree. Consider a vector $d = [d_1^{out}, d_2^{out}, ..., d_n^{out}]^{\mathrm{T}}$, $\mathrm{diag}(d)$ is a square diagonal matrix such that d's entries are on its main diagonal. Adjacency matrix, $A \in \mathbb{R}^{n\times n}$, of the graph $\mathcal{G}$ is defined by

$$[A]_{ij} = k_{ij} \quad \text{for all } (i,j) \in \mathbf{V} \times \mathbf{V}. \tag{1}$$

Laplacian matrix of graph $\mathcal{G}$ is defined as $L = D - A$, where $D = \mathrm{diag}(d)$. Normalized Laplacian matrix of a directed graph is $L_N = DD^{\dagger} - D^{\dagger}A$. Eigenvalues of Laplacian matrix of any directed graph have non-negative real parts [20]. The row sum of Laplacian matrix is equal to zero. Consequently, $L\mathbf{1}_n = 0$ and L has at least one zero eigenvalue. Eigenvalues of Laplacian matrix are shown by λ_i's for all $i \in \mathcal{I}$ and we set $\lambda_1 = 0$. A directed graph has a globally reachable node if and only if it has one zero eigenvalue and the rest of its eigenvalues have positive real parts [22]. For a balanced graph, $\mathbf{1}_n^{\mathrm{T}}L = 0$.

III. Problem Statement

Let us consider a team of mobile robots who want to rendezvous at an appointed location but they do not have any priori knowledge of the gathering time. Therefore, the robots should reach an agreement on their rendezvous time by achieving consensus. We assign a scalar state to each robot i, whose value at time t is denoted by x_i, which is his belief of the gathering time. Moreover, in order to achieve consensus, robots can communicate with each other to share their states. Robots and their directed communication links are represented by nodes and edges of a directed communication graph $\mathcal{G}$, respectively. Direction of the edges represent the direction of communication links, i.e., for an edge (i,j) robot i receives information from robot j. In order to incorporate deficiencies of communication network, we assume all robots experience an identical and constant [1] time-delay, $\tau \in \mathbb{R}^+$. Furthermore, communication noise is modeled by a white Gaussian noise [26]. Dynamics of rendezvous state of robot i, for every $i \in \mathbf{V}$, is given

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

where $u_i(t)$, the control input of robot i, is given by

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

in which k_{ij} is the feedback gain that robot i allocates to robot j and is discussed in equation (1) as an element of adjacency matrix. Output of node i at time t, $y_i(t)$, is defined as deviation of its state from the average, i.e.,

$$y_i(t) := x_i(t) - \frac{1}{n}\sum_{j=1}^{n} x_j(t).$$

States of all nodes at time t are represented by vector $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^{\mathrm{T}}$. Vector of noise $\xi(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_n(t)]^{\mathrm{T}}$ is a Gaussian white noise process with zero mean and identity covariance, i.e., $\mathbb{E}[\xi(t_1), \xi(t_2)] = I_n\delta(t_1 - t_2)$, where $\delta(t)$ is delta function. The dynamics of the whole network can be written as

$$\begin{aligned} \dot{x}(t) &= -Lx(t-\tau) + \xi(t), \quad t \geq 0, \\ y(t) &= M_n x(t), \end{aligned} \tag{2}$$

[1] This assumption has been widely used by other researchers as it allows analytical derivations of formulas. For rendezvous in time, it is a common practice in robotics labs to use identical communication modules for all agents, which results in a uniform communication time-delay. Moreover, in other related applications such as heading alignments, rendezvous in space, and velocity control of agents using a Motion Capture (MoCap) system to observe their spatial locations in indoor labs [23]–[25] all agents experience an identical time-delay to access data through MoCap system.

with initial condition

$$x(t) = \begin{cases} 0 & \text{if } t \in [-\tau, 0) \\ x_0 & \text{if } t = 0 \end{cases}. \tag{3}$$

Matrix L is the Laplacian matrix of the underlying communication graph $\mathcal{G}$ and

$$[L]_{ij} = \begin{cases} k_{i1} + k_{i2} + \cdots + k_{in} & \text{if } i = j, \\ -k_{ij} & \text{otherwise,} \end{cases}$$

Moreover, the output matrix, M_n, is the centering matrix. In our example of rendezvous, we can interpret x_0 in (3) as the preferred rendezvous time of the robots.

Assumption 1. *The communication graph $\mathcal{G}$ has a globally reachable node.*

By the above assumption, $\mathbf{1}_n$ spans the right null space of L. When there is no noise, under Assumption 1, if time-delay is less than some certain threshold, then all robots will achieve consensus. However, in the presence of noise, consensus will not happen. In order to quantify the quality of consensus, we employ the $\mathcal{H}_2$-norm of the network as the performance measure, which is defined as

$$\rho(L, \tau) = \mathbb{E}\left[\int_{t=0}^{\infty} \mathrm{Tr}\left(y(t)^T y(t)\right) dt\right], \tag{4}$$

For network (2), equation (4) measures the dispersion of the states of the robots from the average. In other words, it measures the total amount of disagreement among the robots. As the disagreement increases, the $\mathcal{H}_2$-norm increases. In this paper, we seek a closed-form expression for the $\mathcal{H}_2$-norm of the network (2) when the Laplacian matrix of the communication graph $\mathcal{G}$ is normal. Our expression can be applied to not only undirected communication topologies [16] but also a class of directed ones. Moreover, we suggest an efficient method for computing the $\mathcal{H}_2$-norm of network (2) and its gradient in order to optimize the performance by allocating optimal edge weights to the graph $\mathcal{G}$.

Remark 1. *Let ν to span the left null space of Laplacian matrix. It can be shown that in the absence of noise, the value of consensus is $\frac{\nu x_0}{\nu^{\mathrm{T}} \mathbf{1}_n}$. It is worth mentioning that in section VI, we use the notion of average-consensus. Similar to consensus, in average-consensus all robots want to reach an agreement on their states. However, the value of average-consensus, as its name suggest, is $\frac{\mathbf{1}_n x_0}{n} = \frac{x_1(0)+x_2(0)+\cdots+x_n(0)}{n}$.*

IV. Stability and Convergence Results

To have a meaningful discussion on the performance of reaching consensus, the network has to be stable since otherwise, the $\mathcal{H}_2$-norm will be unbounded. However, in system (2), the system matrix, $-L$, has one zero eigenvalue and as a result it has a marginally stable mode. In the following, we show that how we can transform the system in order to remove the marginally stable mode.

Lemma 1. *Transfer matrix of system (2) is the same as transfer matrix of the following stable modified system*

$$\begin{aligned} \dot{\hat{x}}(t) &= -\tilde{L}\hat{x}(t-\tau) + \Big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}\Big)\xi(t), \quad t \geq 0, \\ y(t) &= M_n \hat{x}(t), \end{aligned} \tag{5}$$

with initial condition

$$\hat{x}(t) = \begin{cases} 0 & \text{for } t \in [-\tau, 0) \\ x_0 - \frac{\nu^{\mathrm{T}} x_0}{\nu^{\mathrm{T}} \mathbf{1}_n} \mathbf{1}_n & \text{for } t = 0 \end{cases}, \tag{6}$$

where vector ν spans left null space of Laplacian matrix L. Also, $\tilde{L} = L + \alpha_d \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}$ and α_d is a positive constant that later on we will decide on its value through a specification. Moreover, the new state variable is $\hat{x}(t) = x(t) - \frac{\nu^{\mathrm{T}} x_0}{\nu^{\mathrm{T}} \mathbf{1}_n} \mathbf{1}_n$.

Proof. Since ν spans left null space of L, in the absence of noise, $\nu^{\mathrm{T}} \dot{x}(t) = 0$. Therefore, $\nu^{\mathrm{T}} x(t)$ is an invariant quantity and $\nu^{\mathrm{T}} x_0 = \nu^{\mathrm{T}} x(t)$. Hence, equivalently $\hat{x}(t) = \big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}\big)x(t)$. Moreover, because $M_n\big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}\big) = M_n$, we get

$$y = M_n \hat{x}(t) = M_n x(t). \tag{7}$$

Transfer matrix of (2) and (5) are

$$\begin{aligned} G_L(s) =& M_n (sI + e^{-s\tau} L)^{-1}, \\ G_{\tilde{L}}(s) =& M_n (sI + e^{-s\tau} \tilde{L})^{-1} \big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}\big), \end{aligned} \tag{8}$$

respectively. The Jordan normal form of matrices L and $\tilde{L}$ are given by

$$\begin{aligned} L =& P \operatorname{diag}([0, J_2, \ldots, J_p]) P^{-1} = P J_L P^{-1}, \\ \tilde{L} =& P \operatorname{diag}([\alpha_d, J_2, \ldots, J_p]) P^{-1} = P J_{\tilde{L}} P^{-1}, \end{aligned} \tag{9}$$

where $\operatorname{diag}([0, J_2, \ldots, J_p])$ is a block diagonal matrix such that 0, J_2, $\ldots$, J_p are its main diagonal blocks, moreover; $\operatorname{diag}([\alpha_d, J_2, \ldots, J_p])$ is similarly defined.
Also, it can be shown that

$$\Big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n}\Big) = P \operatorname{diag}([0, 1, \ldots, 1]) P^{-1} = P \Lambda_T P^{-1}, \tag{10}$$

where $\Lambda_T = \operatorname{diag}([0, 1, \ldots, 1])$. By substituting (9) and (10) into (8) we obtain

$$\begin{aligned} G_L(s) =& M_n P (sI_n + e^{-s\tau} J_L)^{-1} P^{-1}, \\ G_{\tilde{L}}(s) =& M_n P (sI_n + e^{-s\tau} J_{\tilde{L}})^{-1} \Lambda_T P^{-1}. \end{aligned} \tag{11}$$

Note that

$$\begin{aligned} (sI_n + e^{-s\tau} J_L)^{-1} &= \operatorname{diag}([s^{-1}, \hat{J}_2^{-1}, \ldots, \hat{J}_p^{-1}]), \\ (sI_n + e^{-s\tau} J_{\tilde{L}})^{-1} &= \operatorname{diag}([(s + e^{-s\tau}\alpha_d)^{-1}, \hat{J}_2^{-1}, \ldots, \hat{J}_p^{-1}]), \end{aligned} \tag{12}$$

where $\hat{J}_i^{-1} = (sI_{n(i)} + e^{-s\tau} J_i)^{-1}$ for $i \in \{2, 3, \ldots, p\}$ and $n(i)$ is the dimension of the Jordan block J_i. By substituting (12) into (11) we get

$$\begin{aligned} G_L(s) =& M_n P \operatorname{diag}([0, \hat{J}_2^{-1}, \ldots, \hat{J}_p^{-1}]) P^{-1}, \\ G_{\tilde{L}}(s) =& M_n P \operatorname{diag}([0, \hat{J}_2^{-1}, \ldots, \hat{J}_p^{-1}]) P^{-1}. \end{aligned}$$

□

Proposition 1. *The input matrix of the modified system* (5) *is real-valued .*

Proof. We prove this proposition by contradiction. Suppose that ν is not a real vector. Then we can rewrite it as $\nu = \nu_1 + \nu_2 \jmath$, where ν_1 and ν_2 are real and independent vectors. Besides, we have

$$\nu^{\mathrm{T}} L = \nu_1^{\mathrm{T}} L + \nu_2^{\mathrm{T}} L \jmath = 0. \tag{13}$$

Therefore, both real and imaginary parts of (13) should be equal to zero. Consequently, the left null space of $-L$ is equal to the span of ν_1 and ν_2. It is a contradiction since we have assumed that rank of L is $n-1$. As a result, ν is a real vector and input matrix is a real matrix as well. □

In the absence of noise, network (5) is asymptotically stable if and only if all eigenvalues of $\tilde{L}$ have positive real parts, i.e., for all $i \in \mathcal{I}$, $\Re\{\lambda_i\} > 0$, and τ is less than time-delay margin τ^*, i.e., $\tau \in [0, \tau^*)$ [27], where the time-delay margin is

$$\tau^* = \min_i \left(\frac{1}{|\lambda_i|} \arcsin\left(\frac{\Re\{\lambda_i\}}{|\lambda_i|} \right) \right). \tag{14}$$

Since all eigenvalues of $\tilde{L}$ are the same as those of L except the first one, we replace the zero eigenvalue of L with α_d. To preserve the properties of the original network we should have

$$\tau^* \leq \left(\frac{1}{|\alpha_d|} \arcsin\left(\frac{\Re(\alpha_d)}{|\alpha_d|} \right) \right),$$

or equivalently

$$0 < \alpha_d \leq \frac{\pi}{2\tau^*}.$$

Remark 2. *In the absence of noise , if $\tau < \tau^*$ then network* (2) *achieve consensus asymptotically although it is marginally stable.*

V. $\mathcal{H}_2$-NORM EVALUATION

We start by defining $\mathcal{H}_2$-norm of time-delay network (5) and then we restrict our attention to the case where $\tilde{L}$ is a normal matrix and derive a closed-form expression for the $\mathcal{H}_2$-norm.

A. $\mathcal{H}_2$-Norm of LTI Time-Delay Network

The definition of the $\mathcal{H}_2$-norm for the time-delay network is the same as delay-free network. Let $G_{\tilde{L}}$ be the transfer matrix of the system (5), which is equal to the transfer matrix of original system (2), G_L, then by Parceval's theorem, we can rewrite (4) in frequency domain as

$$\frac{1}{2\pi} \int_{-\infty}^{\infty} \mathrm{Tr}\left(G_L(\jmath\omega)^{\mathrm{H}} G_L(\jmath\omega) \right) d\omega. \tag{15}$$

To find an expression for the $\mathcal{H}_2$-norm, in the presence of time-delay, we need to introduce fundamental solution, delay-Lyapunov matrix, and delay-Lyapunov equations.

Definition 1 ([28])**.** *The fundamental solution of the system* (5)*, $K(t)$, is the inverse Laplace transform of $\left(sI_n + \tilde{L}e^{-s\tau}\right)^{-1}$, i.e, $\mathcal{L}[K] = \left(sI_n + \tilde{L}e^{-s\tau}\right)^{-1}$, and satisfies the equation*

$$\frac{d}{dt} K(t) = -\tilde{L} K(t - \tau), \tag{16}$$

for $t \geq 0$, and the initial conditions

$$K(0) = I_n \textit{ and } K(t) = 0 \quad \textit{for} \quad t < 0. \tag{17}$$

It can be shown that the fundamental solution satisfies the following relation, with the same initial condition as (17)

$$\frac{d}{dt} K(t) = -K(t - \tau)\tilde{L}, \tag{18}$$

for $t \geq 0$. Furthermore, since system (5) is exponentially stable, it can be shown that $K(t) \to 0$ exponentially.

Definition 2 ([28])**.** *The time-delay Lyapunov matrix of the system* (5) *is defined as*

$$U(t) := \int_{s=0}^{\infty} K^T(s) M_n^{\mathrm{T}} M_n K(s+t) ds. \tag{19}$$

Since $K(t) = 0$ for $t < 0$ and $K(t) \to 0$ exponentially as $t \to +\infty$, $U(t)$ is well defined.

Furthermore, analogous to the delay-free case we exploit Lyapunov equation to compute $\mathcal{H}_2$-norm [28]. Time-delay Lyapunov equations are

$$\begin{aligned} &U(-t) = U^T(t), && \text{for } t \in [-\tau, \tau] \\ &U^T(\tau)\tilde{L} + \tilde{L}^T U(\tau) = M_n, && \\ &\dot{U}(t) = -U(t-\tau)\tilde{L} && \text{for } t \in [0, \tau]. \end{aligned}$$

Theorem 1 ([12])**.** *The $\mathcal{H}_2$-norm squared of asymptotically stable network* (5) *is*

$$\mathrm{Tr}\left(\left(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n} \right)^{\mathrm{T}} U(0) \left(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}} \mathbf{1}_n} \right) \right). \tag{20}$$

Therefore, we need an explicit expression for $U(0)$ to compute $\mathcal{H}_2$-norm. Let us define matrix $\mathcal{B}$ as

$$\mathcal{B} := \mathbf{M} + \mathbf{N} e^{\mathbf{A}\tau}. \tag{21}$$

where

$$\begin{aligned} &\mathbf{A} := \begin{bmatrix} 0 & -\tilde{L}^{\mathrm{T}} \otimes I_n \\ I_n \otimes \tilde{L}^{\mathrm{T}} & 0 \end{bmatrix}, \\ &\mathbf{M} := \begin{bmatrix} 0 & -\tilde{L}^{\mathrm{T}} \otimes I_n \\ I_{n^2} & 0 \end{bmatrix}, \mathbf{N} := \begin{bmatrix} -I_n \otimes \tilde{L}^{\mathrm{T}} & 0 \\ 0 & -I_{n^2} \end{bmatrix}. \end{aligned} \tag{22}$$

It can be shown that for invertible $\mathcal{B}$, matrix $U(0)$ can be found by [29]

$$\mathrm{vec}\, U(0) = \begin{bmatrix} I_{n^2} & 0_{n^2} \end{bmatrix} \mathcal{B}^{-1} \begin{bmatrix} -\mathrm{vec}\, M_n \\ 0_{n^2 \times 1} \end{bmatrix}. \tag{23}$$

Therefore, by substituting $U(0)$ from (23) into equation (20), we can calculate the value of the $\mathcal{H}_2$-norm squared [29], [30].

B. $\mathcal{H}_2$-norm of Time-delay Network with Normal Laplacian

Normal modified Laplacian matrix $\tilde{L}$ can be written as $\tilde{L} = Q\Lambda_{\tilde{L}}Q^{\mathrm{H}}$, where Q is a unitary matrix and $\Lambda_{\tilde{L}}$ is the diagonal matrix of eigenvalues. In this section, we provide a closed-form expression for the $\mathcal{H}_2$-norm of (2), whenever its modified Laplacian matrix is a normal matrix. It can be shown that if the graph Laplacian matrix, L, is Normal, then the modified Laplacian matrix, $\tilde{L}$, is also normal. Lemma 4 of [13] states that a graph $\mathcal{G}$ with globally reachable node and normal Laplacian matrix is balanced.

Theorem 2. *For system* (2) *with corresponding normal Laplacian matrix, the $\mathcal{H}_2$-norm squared is represented in terms of time-delay and Laplacian matrix eigenvalues as*

$$\frac{1}{2}\sum_{i=2}^{n}\frac{\cos\left(\tau|\lambda_i|\right)}{\Re\{\lambda_i\}-|\lambda_i|\sin\left(\tau|\lambda_i|\right)}. \tag{24}$$

Proof. First, we construct $\mathbf{A}$ for our system. Since $\tilde{L}$ is a normal matrix and $\tilde{L} = Q\Lambda_{\tilde{L}}Q^{\mathrm{H}}$, we have

$$\begin{aligned}\tilde{L}^{\mathrm{H}}\otimes I_n &= (Q\otimes\bar{Q})(\bar{\Lambda}_{\tilde{L}}\otimes I_n)(Q^{\mathrm{H}}\otimes Q^T),\\ I_n\otimes\tilde{L}^{\mathrm{T}} &= (Q\otimes\bar{Q})(I_n\otimes\Lambda_{\tilde{L}})(Q^{\mathrm{H}}\otimes Q^T).\end{aligned} \tag{25}$$

Set matrix $\mathcal{Q}$ and matrix Ω to be as follows

$$\mathcal{Q}=\begin{bmatrix}Q\otimes\bar{Q} & 0_{n^2\times n^2}\\ 0_{n^2\times n^2} & Q\otimes\bar{Q}\end{bmatrix}, \Omega=\begin{bmatrix}0_{n^2\times n^2} & -\bar{\Lambda}_{\tilde{L}}\otimes I_n\\ I_n\otimes\Lambda_{\tilde{L}} & 0_{n^2\times n^2}\end{bmatrix}.$$

Therefore, we can rewrite matrix $\mathbf{A}$ as

$$\mathbf{A}=\mathcal{Q}\Omega\mathcal{Q}^{\mathrm{H}}. \tag{26}$$

Because matrix Q is a unitary matrix, it can be shown that $\mathcal{Q}$ is unitary as well. Also, we have

$$I_{n^2}=(Q\otimes\bar{Q})I_{n^2}(Q^{\mathrm{H}}\otimes Q^T).$$

It follows that

$$\mathbf{M}=\mathcal{Q}\Lambda_{\mathbf{M}}\mathcal{Q}^{\mathrm{H}},\quad \mathbf{N}=\mathcal{Q}\Lambda_{\mathbf{N}}\mathcal{Q}^{\mathrm{H}}, \tag{27}$$

where $\Lambda_{\mathbf{M}}$ and $\Lambda_{\mathbf{N}}$ are

$$\Lambda_{\mathbf{M}}=\begin{bmatrix}0_{n^2\times n^2} & -\Lambda_{\tilde{L}}^{\mathrm{H}}\otimes I_n\\ I_{n^2} & 0_{n^2\times n^2}\end{bmatrix}, \Lambda_{\mathbf{N}}=\begin{bmatrix}-I\otimes\Lambda_{\tilde{L}} & 0\\ 0_{n^2\times n^2} & -I_{n^2}\end{bmatrix}.$$

To construct $\mathcal{B}$, we need to compute $e^{\mathbf{A}\tau}$ first, since $\mathcal{Q}$ is invertible and unitary, i.e., $\mathcal{Q}^{\mathrm{H}}\mathcal{Q}=I_{2n^2}$, we get

$$e^{\mathbf{A}\tau}=\mathcal{Q}e^{\Omega\tau}\mathcal{Q}^{\mathrm{H}}. \tag{28}$$

By substituting (27) and (28) into (21) we have

$$(\mathbf{M}+\mathbf{N}e^{\mathbf{A}\tau})^{-1}=\mathcal{Q}\Big(\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau}\Big)^{-1}\mathcal{Q}^{\mathrm{H}}.$$

As a result, we get the following equivalent expression for equation (23)

$$\operatorname{vec}U(0)=\begin{bmatrix}I_{n^2} & 0_{n^2}\end{bmatrix}\mathcal{Q}(\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})^{-1}\mathcal{Q}^H\begin{bmatrix}-\operatorname{vec}M_n\\ 0\end{bmatrix}. \tag{29}$$

Let us compute $e^{\Omega\tau}$ explicitly. For the scalar function f and a general matrix $M\in\mathbb{C}^{n\times n}$, $f(M)$ has the same dimension as matrix M. Consider a diagonalizable matrix M such that

$$M=P\Sigma P^{-1}=P\ \mathrm{diag}([\sigma_1,\sigma_2,\ldots,\sigma_n])P^{-1},$$

then $f(M)$ is as follows

$$f(M)=Pf(\Sigma)P^{-1}=P\mathrm{diag}([f(\sigma_1),\ldots,f(\sigma_n)])P^{-1}. \tag{30}$$

See [31] for more details about functions of matrices. We infer that M and $f(M)$ have the same eigenvectors. Moreover, we have

$$e^{\Omega\tau}=I_{2n^2}+\Omega\tau+\frac{\Omega^2\tau^2}{2!}+\frac{\Omega^3\tau^3}{3!}+\ldots. \tag{31}$$

It can be shown by induction that the even and odd powers of matrix Ω for $k\in\mathbb{N}$ are given by

$$\begin{aligned}\Omega^{2k}&=(-1)^k\begin{bmatrix}\bar{\Lambda}_{\tilde{L}}^k\otimes\Lambda_{\tilde{L}}^k & 0_{n^2\times n^2}\\ 0_{n^2\times n^2} & \bar{\Lambda}_{\tilde{L}}^k\otimes\Lambda_{\tilde{L}}^k\end{bmatrix},\\ \Omega^{2k-1}&=(-1)^k\begin{bmatrix}0_{n^2\times n^2} & \bar{\Lambda}_{\tilde{L}}^k\otimes\Lambda_{\tilde{L}}^{k-1}\\ -\bar{\Lambda}_{\tilde{L}}^{k-1}\otimes\Lambda_{\tilde{L}}^k & 0_{n^2\times n^2}\end{bmatrix}.\end{aligned} \tag{32}$$

By (30), (31) and (32) we obtain

$$\begin{aligned}(e^{\Omega\tau})_{11}&=\cos\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big),\\ (e^{\Omega\tau})_{12}&=-\bar{\Lambda}_{\tilde{L}}^{\frac{1}{2}}\sin\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big)\Lambda_{\tilde{L}}^{-\frac{1}{2}},\\ (e^{\Omega\tau})_{21}&=\bar{\Lambda}_{\tilde{L}}^{-\frac{1}{2}}\sin\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big)\Lambda_{\tilde{L}}^{\frac{1}{2}},\\ (e^{\Omega\tau})_{22}&=\cos\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big).\end{aligned}$$

It follows that

$$\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau}=\begin{bmatrix}(\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{11} & (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{12}\\ (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{21} & (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{22}\end{bmatrix}.$$

Where $(\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{ij}$ for $i,j\in\{1,2\}$ are given by

$$\begin{aligned}(\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{11}&=-(I_n\otimes\Lambda_{\tilde{L}})(e^{\Omega\tau})_{11}\\ (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{12}&=-(\Lambda_{\tilde{L}}^{\mathrm{H}}\otimes I_n)-(I_n\otimes\Lambda_{\tilde{L}})(e^{\Omega\tau})_{12}\\ (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{21}&=I_{n^2}-(e^{\Omega\tau})_{21}\\ (\Lambda_{\mathbf{M}}+\Lambda_{\mathbf{N}}e^{\Omega\tau})_{22}&=-(e^{\Omega\tau})_{22}\end{aligned} \tag{33}$$

Since matrix $\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)$ is a real matrix, we get

$$\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)^{\mathrm{T}}=\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)^{H}. \tag{34}$$

Recall that by equation (10), we can rewrite $\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)$ as $Q\Lambda_TQ^H$. By substituting (34) in (20) we obtain

$$\begin{aligned}&\mathrm{Tr}\left(\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)^{H}U(0)\left(I_n-\frac{\mathbf{1}_n\nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\right)\right)\\ &\qquad=\mathrm{Tr}\left(Q\Lambda_TQ^HU(0)Q\Lambda_TQ^H\right)\\ &\qquad=\mathrm{Tr}\left(Q\Lambda_TQ^HU(0)\right).\end{aligned} \tag{35}$$

Moreover, for two arbitrary matrices $M\in\mathbb{C}^{n\times n}$ and $\hat{M}\in\mathbb{C}^{n\times n}$ we have

$$\mathrm{Tr}(M^{\mathrm{H}}\hat{M})=\mathrm{vec}(M)^{\mathrm{H}}\,\mathrm{vec}(\hat{M}).$$

Hence, we obtain the following equivalent expression for (35)

$$\mathrm{vec}(Q\Lambda_TQ^H)^T\,\mathrm{vec}(U(0)). \tag{36}$$

By premultiplying equation (29) by $\operatorname{vec}(Q\Lambda_T Q^H)^T$ we obtain

$$\begin{bmatrix}\operatorname{vec}(Q\Lambda_T Q^H)^T & 0\end{bmatrix} \mathfrak{Q}(\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}\mathfrak{Q}^H \begin{bmatrix}-\operatorname{vec} M_n \\ 0\end{bmatrix}$$
$$= \begin{bmatrix}\operatorname{vec}(Q\Lambda_T Q^H)^T (Q\otimes\bar{Q}) & 0\end{bmatrix} (\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}$$
$$\begin{bmatrix}-(Q^H\otimes Q^{\mathrm{T}})\operatorname{vec} {M_n}^T \\ 0\end{bmatrix} = -\operatorname{vec}(\Lambda_T)^{\mathrm{T}}(Q^H\otimes Q^{\mathrm{T}})(Q\otimes\bar{Q})$$
$$\big((\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}\big)_{11}(Q^H\otimes Q^{\mathrm{T}})\operatorname{vec}(\bar{Q}\Lambda_{M_n}Q^{\mathrm{T}})$$
$$= -\operatorname{vec}(\Lambda_T)^{\mathrm{T}}\big((\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}\big)_{11}\operatorname{vec}(\Lambda_{M_n}).$$

For matrix $P = \begin{bmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{bmatrix}$ which consists of 4 blocks, if P_{22} is non-singular, and the Schur complement of block P_{22}, i.e., $P_{11} - P_{12}P_{22}^{-1}P_{21}$, is invertible, then the first entry of the inverse of the block matrix is given by [32]

$$(P^{-1})_{11} = (P_{11} - P_{12}P_{22}^{-1}P_{21})^{-1}. \tag{37}$$

Hence, by substituting from (33) into (37) we get

$$\big((\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}\big)_{11} = -\Bigg((I_n\otimes\Lambda_{\tilde{L}})\cos\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big)$$
$$+\big[(\bar{\Lambda}_{\tilde{L}}\otimes I_n) - (I_n\otimes\Lambda_{\tilde{L}})\bar{\Lambda}_{\tilde{L}}^{\frac{1}{2}}\sin\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big){\Lambda_{\tilde{L}}}^{-\frac{1}{2}}\big]$$
$$\Big(\cos\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big)\Big)^{-1}\big[I_{n^2} - \bar{\Lambda}_{\tilde{L}}^{-\frac{1}{2}}\sin\big((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau\big){\Lambda_{\tilde{L}}}^{\frac{1}{2}}\big]\Bigg)^{-1}.$$

Since $\Lambda_{\tilde{L}}$ and $\bar{\Lambda}_{\tilde{L}}$ are diagonal matrices, their kronecker product, i.e., $(\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})$ is a diagonal matrix as well, hence $\sin((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau)$ and $\cos((\bar{\Lambda}_{\tilde{L}}\otimes\Lambda_{\tilde{L}})^{\frac{1}{2}}\tau)$ is a diagonal matrix which its diagonal elements are sine and cosine of the real numbers, thus we do the computation for all the diagonal elements of the matrix $\big((\Lambda_{\mathbf{M}} + \Lambda_{\mathbf{N}} e^{\Omega\tau})^{-1}\big)_{11}$ and then after multiplying it by $\operatorname{vec}(\Lambda_{M_n})$ from right and by $(-\operatorname{vec}(\Lambda_T)^T)$ from left we get the equation (24). □

For delay-free case, equation (24) is simplified to

$$\frac{1}{2}\sum_{i=2}^{n}\frac{1}{\Re\{\lambda_i\}},$$

whichi is reported in [13]. Moreover, for undirected graphs, Laplacian matrix is symmetric and $\Re\{\lambda_i\} = \lambda_i = |\lambda_i|$. Therefore, by substituting in (24) we obtain

$$\frac{1}{2}\sum_{i=2}^{n}\frac{\cos(\tau\lambda_i)}{\lambda_i\big(1-\sin(\tau\lambda_i)\big)},$$

which is the expression that is given in [16], [33] for time-delay undirected networks. In the rest of this section, we study the behavior of (24) as a function of time-delay.

Theorem 3. *For network* (2) *with a normal Laplacian matrix, the $\mathcal{H}_2$-norm squared is an increasing function of time-delay for all $\tau < \tau^*$.*

Proof. By taking derivative of (24) with respect to time-delay (τ) we get the following

$$\frac{1}{2}\sum_{i=2}^{n}\frac{|\lambda_i|^2 - \sin(\tau|\lambda_i|)|\lambda_i|\,\Re(\lambda_k)}{\Big(\Re(\lambda_i) - |\lambda_i|\sin(\tau(|\lambda_i|))\Big)^2}. \tag{38}$$

Our system is exponentially stable, therefore the denominator of equation (38) is positive and for every $i\in\{2,...n\}$ we have

$$|\lambda_i|\sin(\tau|\lambda_i|) < \Re(\lambda_i). \tag{39}$$

Multiplying both sides of the (39) by $(-\Re(\lambda_i))$ we get

$$-|\lambda_i|\,\Re(\lambda_i)\sin(\tau|\lambda_i|) > -(\Re(\lambda_i))^2. \tag{40}$$

Add $|\lambda_i|^2$ to the both sides of the equation (40)

$$|\lambda_i|^2 - \sin(\tau|\lambda_i|)|\lambda_i|\,\Re(\lambda_i) > |\lambda_i|^2 - (\Re(\lambda_i))^2,$$

since $|\lambda_i|^2 - (\Re(\lambda_i))^2 \ge 0$ we obtain

$$|\lambda_i|^2 - \sin(\tau|\lambda_i|)|\lambda_i|\,\Re(\lambda_i) > 0.$$

Hence, the numerator of (38) is also positive and thus, Theorem 3 follows. □

Theorem 4. *If the corresponding Laplacian matrix of network* (2) *is normal, then among all communication graphs on n nodes with normal Laplacian, complete graph reaches the minimum achievable value of the spectral function* (24)*, which is equal to*

$$\rho_n^* \approx 1.5319(n-1)\tau. \tag{41}$$

Remark 3. *According to our numerical results, we suggest a conjecture which is as follows: The $\mathcal{H}_2$-norm squared of asymptotically stable time-delay network* (2) *is lower-bounded by that of an asymptotically stable time-delay network with normal Laplacian matrix and the same eigenvalues, i.e.,* $\frac{1}{2}\sum_{i=2}^{n}\frac{\cos\big(\tau|\lambda_i|\big)}{\Re\{\lambda_i\}-|\lambda_i|\sin\big(\tau|\lambda_i|\big)}$.

Figure 1 illustrates an empirical study. Horizontal axis indicates number of nodes and the vertical axis represents the square of the ratio of the $\mathcal{H}_2$-norm of the two networks. For a fixed number of nodes, we collect and depict results of 100 different sample networks. Moreover, in each sample, time-delay is a random number between zero and the corresponding time-delay margin. One can observe that 1 is a hard limit for all these relative performance measures. It is worth mentioning, this result is valid for the delay-free average-consenus networks and is reported in [34].

C. Scaling Properties of $\mathcal{H}_2$-norm for Families of Graphs

In this section, we discuss the $\mathcal{H}_2$-norm of star graphs, complete graphs, cycle graphs, and path graphs. This discussion is an extension of the case studies reported in [13] for delay-free case. As we mentioned earlier, if the graph Laplacian matrix, L, is Normal, then the modified Laplacian matrix, $\tilde{L}$, is also normal.

Directed star graph $\mathcal{S}_n$: Imploding star graph on n nodes and $n-1$ edges has a central node such that any other node is connected to it by an edge. All edges of the graph points towards the central node. We set edge weights to 1. Then, Laplacian matrix of star graph is

$$L_{\mathcal{S}_n} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0\\ -1 & 1 & \dots & 0 & 0\\ \vdots & \vdots & \ddots & 0 & 0\\ -1 & 0 & \dots & 1 & 0\\ -1 & 0 & \dots & 0 & 1\end{bmatrix}.$$

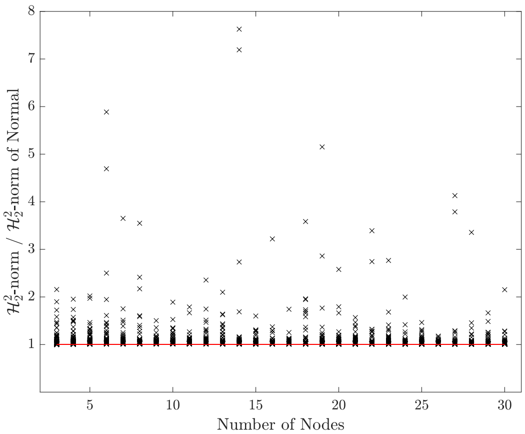

Fig. 1. Lower bound on the $\mathcal{H}_2$-norm of consensus networks

Eigenvalue of $L_{\mathcal{S}_n}$ are $\{0, 1, \dots, 1\}$. As a result, from (14) we obtain the corresponding time-delay margin $\tau^* = \frac{\pi}{2}$. It can be shown that $L_{\mathcal{S}_n}$ is not a normal matrix. But if we choose $\alpha_d = 1$, then the modified Laplacian $\tilde{L}_{\mathcal{S}_n}$ is normal. From (24) we obtain the $\mathcal{H}_2$-norm squared

$$\rho(L_{\mathcal{S}_n}, \tau) = \frac{(n-1)}{2} \tan\left(\frac{\pi}{4} + \frac{\tau}{2}\right).$$

Undirected star graph $\tilde{\mathcal{S}}_n$: Undirected star graph on n nodes and $n-1$ edges consists of a central node that connects to all other nodes by undirected edges. We set edge weights to $\frac{1}{2}$. Then, Laplacian matrix of this graph is

$$L_{\tilde{\mathcal{S}}_n} = \begin{bmatrix} \frac{n-1}{2} & -\frac{1}{2} & \dots & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ -\frac{1}{2} & 0 & \dots & \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & \dots & 0 & \frac{1}{2} \end{bmatrix}.$$

Eigenvalues of $L_{\tilde{\mathcal{S}}_n}$ are $\{0, \frac{1}{2}, \dots, \frac{1}{2}, \frac{n}{2}\}$. As a result, from (14), we obtain the correcponding time-delay margin $\tau^*(n) = \frac{\pi}{n}$, which is a decreasing function of number of nodes. Since $\tilde{L}_{\tilde{\mathcal{S}}_n}$ is symmetric, the $\mathcal{H}_2$-norm squared is

$$\rho(L_{\tilde{\mathcal{S}}_n}, \tau) = (n-2) \tan\left(\frac{\tau + \pi}{4}\right) + \frac{1}{n} \tan\left(\frac{\tau n + \pi}{4}\right).$$

Complete graphs $\mathcal{K}_n$: Complete graph is an undirected graph on n nodes that has $\frac{n(n-1)}{2}$ edges, such that there is an undirected edge between any pair of its nodes. We set edge weights to 1. Laplacian matrix of this graph is equal to $nI_n - \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}$. However, we compute the $\mathcal{H}_2$-norm squared of the normalized Laplacian matrix to make our comparison reasonable. Normalized Laplacian matrix is

$$L_{\mathcal{K}_n} = \frac{n}{n-1} I_n - \frac{1}{n-1} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}.$$

Eigenvalues of $L_{\mathcal{K}_n}$ are $\{0, \frac{n}{n-1}, \dots, \frac{n}{n-1}\}$. As a result, from (14), we obtain the corresponding the time-delay margin $\tau^*(n) = \frac{\pi}{2} \frac{n-1}{n}$, which is a decreasing function of number of nodes. Since $L_{\mathcal{K}_n}$ is symmetric and normal, $\mathcal{H}_2$-norm squared is

$$\rho(L_{\mathcal{K}_n}, \tau) = \frac{(n-1)^2}{2n} \tan\left(\frac{\pi}{4} + \frac{\tau}{2} \frac{n}{n-1}\right).$$

Directed cycle graphs $\mathcal{C}_n$: Directed cycle graph on n nodes consists of one cycle with n edges that are oriented in the same direction. We set edge weights to 1. Laplacian matrix of this graph is

$$L_{\mathcal{C}_n} = I_n - \begin{bmatrix} \mathbf{e}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{e}_n^{\mathrm{T}} \\ \mathbf{e}_1^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \\ -1 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

Eigenvalues of $L_{\mathcal{C}_n}$ are

$$\lambda_{i+1} = 1 + e^{\jmath \pi (1 - \frac{2i}{n})} \quad \text{for } i \in \{0, 1, \dots, n-1\}.$$

Therefore, we have

$$\Re(\lambda_{\mathcal{C}_n, i}) = 2 \sin^2\left(\frac{\pi i}{n}\right), |\lambda_{\mathcal{C}_n, i}| = 2 \sin\left(\frac{\pi i}{n}\right). \tag{42}$$

As a result, from (14), we obtain the corresponding time-delay margin $\tau^*(n) = \frac{\pi}{2n \sin(\frac{\pi}{n})}$, which is a decreasing function of n. Hence, $\lim_{n \to \infty} \tau^*(n) = \frac{1}{2}$. It can be shown that $L_{\mathcal{C}_n}$ is a normal matrix. Therefore, by (24) we get

$$\rho(L_{\mathcal{C}_n}, \tau) = \frac{1}{4} \sum_{i=1}^{n-1} \frac{\cos(2\tau \sin(\frac{\pi i}{n}))}{\sin(\frac{\pi i}{n}) \Big(\sin(\frac{\pi i}{n}) - \sin\big(2\tau \sin(\frac{\pi i}{n})\big) \Big)}.$$

In order to find an approximation of the $\mathcal{H}_2$-norm squared for $0 < x < \pi$, we define function $f(x)$ as

$$f(x) := \frac{\cos(2\tau \sin(x))}{\sin(x) \Big(\sin(x) - \sin\big(2\tau \sin(x)\big) \Big)}.$$

Now set

$$\begin{aligned} f_1(x) &= \lim_{x \to 0^+} f(x) = \frac{1}{1 - 2\tau} \frac{1}{x^2}, \\ f_2(x) &= \lim_{x \to \pi^-} f(x) = \frac{1}{1 - 2\tau} \frac{1}{(x - \pi)^2}. \end{aligned} \tag{43}$$

$$\begin{aligned} f_1\left(\frac{\pi}{2}\right) + f_2\left(\frac{\pi}{2}\right) &= \frac{1}{1 - 2\tau} \frac{8}{\pi^2}, \\ f\left(\frac{\pi}{2}\right) &= \tan\left(\frac{\pi}{4} + \tau\right). \end{aligned} \tag{44}$$

From (43) and (44), we obtain

$$f_{\text{approx.}}(x) = \frac{1}{1 - 2\tau} \Big(\frac{1}{x^2} + \frac{1}{(x - \pi)^2} - \frac{8}{\pi^2} \Big) + \tan\left(\frac{\pi}{4} + \tau\right), \tag{45}$$

where $f_{\text{approx.}}(x)$ is the approximation of function $f(x)$. Set $x = \frac{\pi i}{n}$ in equation (45). We get the following approximation for the $\mathcal{H}_2$-norm squared of the cycle graph

$$\frac{n^2}{2\pi^2 (1 - 2\tau)} \sum_{i=1}^{n-1} \frac{1}{i^2} + \frac{n-1}{4} \Big(\tan\left(\frac{\pi}{4} + \tau\right) - \frac{8}{\pi^2 (1 - 2\tau)} \Big), \tag{46}$$

where $\sum_{i=1}^{n-1} \frac{1}{i^2}$ is a generalized harmonic number. It is upper bounded by Riemann zeta function of 2, i.e.,

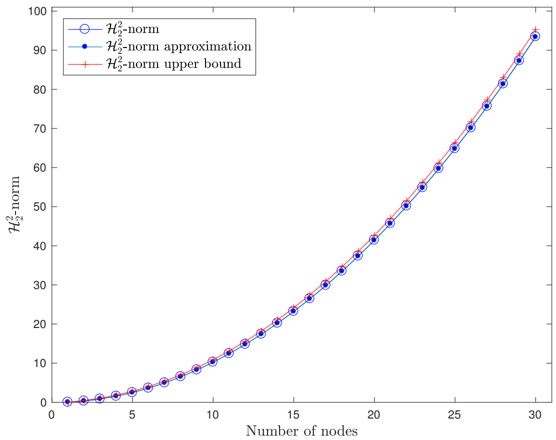

Fig. 2. $\mathcal{H}_2$-norm squared, its approximation and its upper-bound in directed cycle graphs with 0.1 time-delay

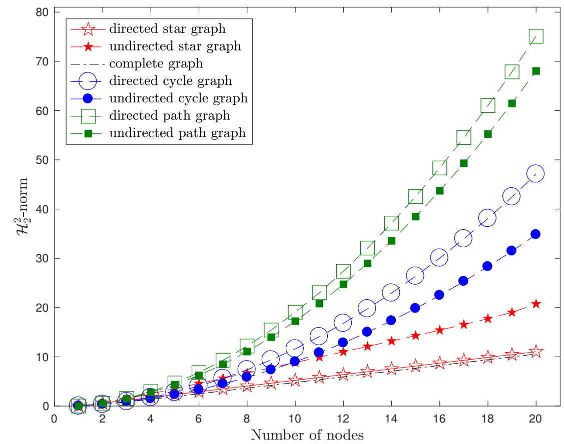

Fig. 3. Comparison between $\mathcal{H}_2$-norm squared of families of graphs with $\tau = \frac{\pi}{21}$

$\sum_{i=1}^{n-1} \frac{1}{i^2} < \zeta(2) = \frac{\pi^2}{6}$. Consequently, an upper bound for equation (46) is

$$\frac{\pi^2 n^2}{12\pi^2(1-2\tau)} + \frac{n-1}{4}\left(\tan(\frac{\pi}{4}+\tau) - \frac{8}{\pi^2(1-2\tau)}\right).$$

Figure 2, illustrates $\mathcal{H}_2$-norm squared of the directed cycle graph, its approximation and the upper-bound on the approximation.

Undirected cycle graphs $\tilde{\mathcal{C}}_n$: Undirected cycle graph on n nodes consists of one cycle with n edges. We set edge weights to $\frac{1}{2}$. Laplacian matrix of this graph is

$$L_{\tilde{\mathcal{C}}_n} = \frac{L_{\mathcal{C}_n} + L_{\mathcal{C}_n}^{\mathrm{T}}}{2}, \tag{47}$$

which is the symmetric part of $L_{\mathcal{C}_n}$. Eigenvalues of $L_{\tilde{\mathcal{C}}_n}$ are

$$\lambda_{i+1} = 1 - \cos\left(\frac{2\pi i}{n}\right) \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

As a result, from (14), we obtain the corresponding time-delay margin

$$\tau^* = \begin{cases} \frac{\pi}{4}, & \text{for even } n, \\ \frac{\pi}{2}\frac{1}{1+\cos(\frac{\pi}{n})}, & \text{for odd } n. \end{cases}$$

The $\mathcal{H}_2$-norm squared is

$$\rho(L_{\tilde{\mathcal{C}}_n}, \tau) = \frac{1}{4}\sum_{i=1}^{n-1} \frac{\tan(\frac{\pi}{4} + \tau\sin^2(\frac{\pi i}{n}))}{\sin^2(\frac{\pi i}{n})}.$$

Directed path graph $\mathcal{P}_n$: If we omit one edge from a directed cycle graph on n nodes, we get a directed path graph on n nodes that has $n-1$ edges. We set edge weights to 1. Laplacian matrix of this graph is

$$L_{\mathcal{P}_n} = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}.$$

Eigenvalues of $L_{\mathcal{P}_n}$ are $\{0, 1, \ldots, 1\}$. As a result, from (14), we obtain the corresponding time-delay margin $\tau^* = \frac{\pi}{2}$. It can be shown that $L_{\mathcal{P}_n}$ is not normal, therefore, we cannot apply formula (24) to compute $\mathcal{H}_2$-norm squared.

Undirected path graph $\tilde{\mathcal{P}}_n$: If we omit directions from a directed path graph, $\mathcal{P}_n$, we get an undirected path graph, $\tilde{\mathcal{P}}_n$, with n nodes and $n-1$ edges. We set edge weights to $\frac{1}{2}$. Laplacian matrix of this graph is

$$L_{\tilde{\mathcal{P}}_n} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & \dots & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & \dots & 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Eigenvalues of $L_{\tilde{\mathcal{P}}_n}$ are

$$\lambda_{i+1} = 2\sin^2\left(\frac{\pi i}{2n}\right), \quad \text{for } i \in \{0, 1, \ldots, n-1\}.$$

As a result, from (14), we obtain the corresponding time-delay margin, $\tau^*(n) = \frac{\pi}{4\cos^2(\frac{\pi}{2n})}$, which is a function of number of nodes. Since $L_{\tilde{\mathcal{P}}_n}$ is symmetric, the $\mathcal{H}_2$-norm squared is

$$\rho(L_{\tilde{\mathcal{P}}}, \tau) = \frac{1}{4}\sum_{i=1}^{n-1} \frac{\tan(\frac{\pi}{4} + \tau\sin^2(\frac{\pi i}{2n}))}{\sin^2(\frac{\pi i}{2n})}.$$

In Figure 3, we illustrate the performance of the aforementioned graphs on 20 nodes with $\tau = \frac{\pi}{21}$. In delay-free case, the $\mathcal{H}_2$-norm squared of directed and undirected cycle graphs are the same. Furthermore, numerical results show that the $\mathcal{H}_2$-norm squared of directed and undirected path graphs are the same too [13]. However, in presence of time-delay this result is not valid anymore and undirected cycle and path graphs have a superior performance with respect to directed counerparts as we can see in Figure 3. Moreover, directed star graph out-performs undirected star graph like delay-free case. Complete graph has the best performance and directed star graph approaches to it similar to delay-free case [13].

VI. Designing Time-Delay Directed Networks with Fixed Topologies

Let us consider the problem of enhancing the performance of a time-delay directed network in achieving average-consensus. Suppose that the graph topology and time-delay τ are fixed. Therefore, we can adjust the edge weights in order to improve the performance. We can cast this problem as a constrained optimization problem. The objective function is minimizing $\rho(L,\tau)$. Edge weights (i.e., feedback gains) are the design parameters and they should be designed such that: (i) they are non-negative, (ii) the resultaning time-delay network remains stable, and (iii) network's graph is balanced in order to reach average-consensus, i.e., each agent tracks the average of states of all nodes .

The class of Laplacian matrices that satisfies (iii) includes symmetric and normal matrices as its proper subset. Although we impose constraint (iii), we can remove it and relax the optimization problem when we simply require consensus.

Let us recall equation (23) which is

$$\begin{aligned}\operatorname{vec} U(0) &= \begin{bmatrix} I_{n^2} & 0_{n^2} \end{bmatrix} \mathcal{B}^{-1} \begin{bmatrix} -\operatorname{vec} M_n \\ 0_{n^2\times 1} \end{bmatrix} &(48)\\ &= \begin{bmatrix} I_{n^2} & 0_{n^2} \end{bmatrix} (\mathbf{M} + \mathbf{N} e^{\mathbf{A}\tau})^{-1} \begin{bmatrix} -\operatorname{vec} M_n \\ 0_{n^2\times 1} \end{bmatrix}. &(49)\end{aligned}$$

Therefore, from (20) we obtain

$$\begin{aligned}\rho(L,\tau) &= \operatorname{Tr}\Big(\big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\big)\big(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n}\big)^{\mathrm{T}} U(0)\Big) \\ &= \Big(\operatorname{vec}\big((I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n})^{\mathrm{T}}(I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n})\big)\Big)^{\mathrm{T}} \operatorname{vec} U(0).\end{aligned}$$

Let matrix B to be equal to the input matrix, i.e., $B = (I_n - \frac{\mathbf{1}_n \nu^{\mathrm{T}}}{\nu^{\mathrm{T}}\mathbf{1}_n})$. By substituting (48) in the equation above we get

$$\rho(L,\tau) = \begin{bmatrix} \operatorname{vec}(BB^{\mathrm{T}}) \\ 0_{n^2\times 1} \end{bmatrix}^{\mathrm{T}} (\mathbf{M} + \mathbf{N} e^{\mathbf{A}\tau})^{-1} \begin{bmatrix} -\operatorname{vec} M_n \\ 0_{n^2\times 1} \end{bmatrix}. \quad (50)$$

Hence, the optimization problem is as follows and the first three constraints are taking care of the three design requirements (i), (ii), and (iii).

$$\begin{aligned} \underset{w_{ij}\forall (i,j)\in \mathbf{E}}{\text{minimize}} \quad & \rho(L,\tau) \\ \text{subject to:} \quad & \forall i \in \mathcal{I} : \tau - \frac{1}{|\lambda_i|}\arcsin\left(\frac{\Re\{\lambda_i\}}{|\lambda_i|}\right) \le 0 \\ & \forall (i,j) \in \mathbf{E} : w_{ij} \ge 0 \\ & \mathbf{1}_n^{\mathrm{T}} L = 0 \\ & L = \sum_{(i,j)\in \mathbf{E}} w_{ij} \mathbf{e}_i (\mathbf{e}_i - \mathbf{e}_j)^{\mathrm{T}}. \end{aligned} \quad (51)$$

The last constraint highlights role of L as a function of decision variable in the cost function. We can get rid of the nonlinear inequality constraints (i.e., the first constraint) by setting objective function to be equal to infinity whenever nonlinear inequality is violated. We use Interior Point OPTimizer (IPOPT) solver to solve our optimization problem [35]. As we use quasi-Newton based methods for low-cost computing (updating) Hessian in each iteration of the optimization process, one can verify that objective function and gradient evaluations are the most expensive (time consuming) steps. Herein, we outline our method for computing objective function and its gradient efficiently to reduce time-complexity.

A. Objective Function Computation

In order to accelerate computation of objective function as in equation (50), we employ the structure of matrices in our formulation. We start with computing matrix exponential $e^{\mathbf{A}\tau}$. This matrix exponential is computed via scaling and squaring method, along with Padé approximation, which is the most widely used method for computing matrix exponential [36]. We briefly introduce the Padé approximants and scaling and squaring method [37]. For a square matrix M, let us define $\mathcal{N}_{p,q}(M)$ and $\mathcal{D}_{p,q}(M)$ as

$$\begin{aligned}\mathcal{N}_{p,q}(M) &= \sum_{j=0}^{p} \frac{(p+q-j)!p!}{(p+q)!j!(p-j)!} M^j, \\ \mathcal{D}_{p,q}(M) &= \sum_{j=0}^{q} \frac{(p+q-j)!q!}{(p+q)!j!(q-j)!} (-M)^j.\end{aligned}$$

Then, the $[p/q]$ Padé approximants of matrix M is defined by

$$\mathcal{R}_{p,q}(M) \coloneqq \mathcal{D}_{p,q}(M)^{-1}\,\mathcal{N}_{p,q}(M).$$

The accuracy of Padé approximants depend on the subordinate norm of matrix M. If the subordinate norm of matrix M is in order of 1, then $\mathcal{R}_{p,q}(M)$ approximates e^M with small error, i.e., $\mathcal{R}_{p,q}(M) \approx e^M$ [36]. Scaling and squaring method suggests dividing matrix M by integral powers of 2 such that subordinate norm of $\frac{M}{2^s}$ is in order of 1. Our numerical results shows that for $\mathbf{A}\tau$, s is equal to zero. In other words, we can skip scaling and squaring step in computing $e^{\mathbf{A}\tau}$, i.e., $\mathcal{R}_{p,q}(e^{\mathbf{A}\tau}) \approx e^{\mathbf{A}\tau}$. Let $\mathcal{D}$ and $\mathcal{N}$ denote the shorthand notations for $\mathcal{D}_{p,q}(\mathbf{A}\tau)$ and $\mathcal{N}_{p,q}(\mathbf{A}\tau)$, respectively. Hence, the objective function (50) can be written as

$$\rho(L,\tau) = \begin{bmatrix} \operatorname{vec}(BB^{\mathrm{T}}) \\ 0_{n^2\times 1} \end{bmatrix}^{\mathrm{T}} \mathcal{D}(\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N})^{-1} \begin{bmatrix} -\operatorname{vec} M_n \\ 0_{n^2\times 1} \end{bmatrix}.$$

Now, if

$$\begin{aligned} z_1^{\mathrm{T}} &= \begin{bmatrix} \operatorname{vec}(BB^{\mathrm{T}})^{\mathrm{T}} & 0_{1\times n^2} \end{bmatrix} \mathcal{D}, \\ z_2 &= (\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N})^{-1} \begin{bmatrix} -\operatorname{vec} M_n \\ 0_{n^2\times 1} \end{bmatrix}, \end{aligned} \quad (52)$$

then $\rho(L,\tau) = z_1^{\mathrm{T}} z_2$.

Let us remark that $\mathcal{R}_{i,i}$, where $i = \max(p,q)$, has higher accuracy than $\mathcal{R}_{p,q}$ at the same cost. Therefore, we exploit $\mathcal{R}_{i,i}$ as Padé approximants [36]. In our implementation, we exploit $[13/13]$ Padé approximants, i.e., $p = q = 13$.

In order to compute $\mathcal{R}_{p,q}(e^{\mathbf{A}\tau})$, we should construct matrix $\mathcal{D}$ and $\mathcal{N}$. Define V_1 and V_2 as

$$\begin{aligned} V_1 \coloneqq& \mathbf{A}\tau[\mathbf{A}^6\tau^6(\beta_{13}\mathbf{A}^6\tau^6 + \beta_{11}\mathbf{A}^4\tau^4 + \beta_9\mathbf{A}^2\tau^2) \\ &+ \beta_7\mathbf{A}^6\tau^6 + \beta_5\mathbf{A}^4\tau^4 + \beta_3\mathbf{A}^2\tau^2 + \beta_1 I_{2n^2}], \\ V_2 \coloneqq& \mathbf{A}^6\tau^6(\beta_{12}\mathbf{A}^6\tau^6 + \beta_{10}\mathbf{A}^4\tau^4 + \beta_8\mathbf{A}^2\tau^2) \\ &+ \beta_6\mathbf{A}^6\tau^6 + \beta_4\mathbf{A}^4\tau^4 + \beta_2\mathbf{A}^2\tau^2 + \beta_0 I_{2n^2}, \end{aligned}$$

where β_i's are the coefficients of $[13/13]$ Padé approximants that are as follows

$$\begin{array}{ll}\beta_{13} = 64764752532480000 & \beta_{12} = 32382376266240000\\ \beta_{11} = 7771770303897600 & \beta_{10} = 1187353796428800\\ \beta_9 = 129060195264000 & \beta_8 = 10559470521600\\ \beta_7 = 670442572800 & \beta_6 = 33522128640\\ \beta_5 = 1323241920 & \beta_4 = 40840800\\ \beta_3 = 960960 & \beta_2 = 16380\\ \beta_1 = 182 & \beta_0 = 1.\end{array}$$

Hence, we obtain the following equivalent expressions for $\mathcal{N}$ and $\mathcal{D}$

$$\mathcal{N} = V_1 + V_2, \quad \mathcal{D} = V_2 - V_1,$$

Define an arbitrary augmented vector $u \in \mathbb{R}^{2n^2}$ as

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{53}$$

where u_1 and u_2 are in $\mathbb{R}^{n^2}$ and U_1 and U_2 are the corresponding matrices in $\mathbb{R}^{n\times n}$ such that $\mathrm{vec}(U_1) = u_1$ and $\mathrm{vec}(U_2) = u_2$. By (22) for $n \geq 1$ we obtain

$$\begin{aligned} \mathbf{M}u &= \begin{bmatrix} -\mathrm{vec}(U_2\tilde{L}) \\ u_1 \end{bmatrix}, \quad \mathbf{N}u = -\begin{bmatrix} \mathrm{vec}(\tilde{L}^{\mathrm{T}}U_1) \\ u_2 \end{bmatrix}, \\ \mathbf{A}^{2n}u &= (-1)^n \begin{bmatrix} \mathrm{vec}\left((\tilde{L}^n)^{\mathrm{T}}U_1\tilde{L}^n\right) \\ \mathrm{vec}\left((\tilde{L}^n)^{\mathrm{T}}U_2\tilde{L}^n\right) \end{bmatrix}, \\ \mathbf{A}^{2n-1}u &= (-1)^n \begin{bmatrix} \mathrm{vec}\left((\tilde{L}^{n-1})^{\mathrm{T}}U_2\tilde{L}^n\right) \\ -\mathrm{vec}\left((\tilde{L}^n)^{\mathrm{T}}U_1\tilde{L}^{n-1}\right) \end{bmatrix}, \end{aligned} \tag{54}$$

under the convention that $\tilde{L}^0 = I_n$. Therefore, time complexity of computing $\mathcal{N}u$ and $\mathcal{D}u$ is $\mathcal{O}(n^3)$.

Now we have all the tools that we need in order to compute $\rho(L,\tau) = z_1^{\mathrm{T}} z_2$. We start by computing z_2. Recall that $z_2 = (\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N})^{-1}\begin{bmatrix} -\mathrm{vec}\,M_n \\ 0_{n^2\times 1} \end{bmatrix}$. To avoid computing matrix inverse for finding z_2 with time complexity $\mathcal{O}(n^6)$, one can think of finding z_2 as solving

$$(\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N})z_2 = \begin{bmatrix} -\mathrm{vec}\,M_n \\ 0_{n^2\times 1} \end{bmatrix}$$

for z_2. This equation can be solved by GMRES algorithm efficiently [38]. GMRES algorithm is an iterative method that contains only matrix-vector multiplications and as a result, it is a fast method. According to equations (54), time complexity of finding product of $\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N}$ and a vector u is $\mathcal{O}(n^3)$. However, if matrix $\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N}$ has a large condition number, iterative methods (like GMRES) cannot converge fast. Robustness and convergence rate of iterative methods can be increased by a suitable preconditioner [39]. A preconditioner $\mathcal{P}$ should be chosen such that $\mathcal{P}^{-1}u$ can be done very fast, where u is a vector. Since for $\tau = 0$, $\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N}$ equals $\beta_0(\mathbf{M} + \mathbf{N})$, one can employ $\mathbf{M} + \mathbf{N}$ as a left preconditioner. When time-delay is zero, $\mathbf{M} + \mathbf{N}$ is a perfect preconditioner since the condition number of $(\mathbf{M} + \mathbf{N})^{-1}(\mathbf{M}\,\mathcal{D} + \mathbf{N}\,\mathcal{N})$ is one. Let $T = \left(I_n \otimes (-\tilde{L})^{\mathrm{T}}\right) + \left(-\tilde{L}^{\mathrm{T}} \otimes I_n\right)$, then inverse of $\mathbf{M} + \mathbf{N}$ is

$$\mathcal{P}^{-1} = \begin{bmatrix} T^{-1} & T^{-1}(-\tilde{L}^{\mathrm{T}} \otimes I_n) \\ T^{-1} & -I_{n^2} + T^{-1}(-\tilde{L}^{\mathrm{T}} \otimes I_n) \end{bmatrix}.$$

Consider vector u as (53), then we have

$$\mathcal{P}^{-1}u = \begin{bmatrix} T^{-1}u_1 - T^{-1}\,\mathrm{vec}(U_2\tilde{L}) \\ T^{-1}u_1 - T^{-1}\,\mathrm{vec}(U_2\tilde{L}) - u_2 \end{bmatrix}.$$

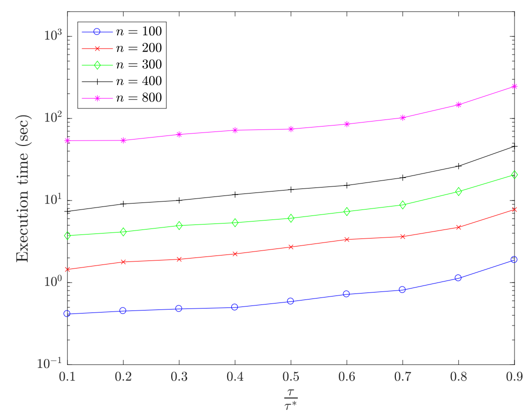

Fig. 4. Execution time (sec) for computing the performance for different network sizes. We generated a graph with random topology and weights. For each network size, the graphs are common and only time-delay is varying.

TABLE I
NUMBER OF REQUIRED GMRES ITERATIONS TO FIND THE PERFORMANCE.

n \ $\frac{\tau}{\tau^*}$	0.1	0.3	0.5	0.7	0.9
10	9	14	20	28	44
50	9	13	19	29	62
100	8	12	16	24	50
200	8	11	15	22	46
500	7	10	14	19	45
1000	7	9	12	17	37

Therefore, efficient matrix-vector multiplication for $\mathcal{P}^{-1}$ depends on the efficiency of matrix-vector multiplication for matrix T^{-1}. Matrix-vector multiplication for matrix T^{-1} can be done efficiently in $\mathcal{O}(n^3)$ by solving delay-free Lyapunov equation. As an example, let $\mathrm{vec}(X) = T^{-1}u_1$, then we have

$$\begin{aligned} u_1 &= T\,\mathrm{vec}(X) \\ &= \Big(\big(I_n \otimes (-\tilde{L})^{\mathrm{T}}\big) + \big(-\tilde{L}^{\mathrm{T}} \otimes I_n\big)\Big)\,\mathrm{vec}(X) \\ &= \mathrm{vec}(-\tilde{L}^{\mathrm{T}}X) + \mathrm{vec}(-X\tilde{L}), \end{aligned}$$

which is equivalent to

$$\tilde{L}^{\mathrm{T}}X + X\tilde{L} + U_1 = 0, \tag{55}$$

where the above equation is delay-free Lyapunov equation. Hence, time-complexity of calculating $\mathcal{P}^{-1}u$ is $\mathcal{O}(n^3)$. We solve Lyapunov equation through MATLAB's *lyap* function. Since in most of the iterations, matrix U_1 in equation (55) is not symmetric, MATLAB calls SLICOT subroutines to solve it as a Sylvester equation. These subroutines, use a Hessenberg-Schur method to solve the Sylvester equations [40]. In order to compute z_2, we use GMRES algorithm with $\mathbf{M} + \mathbf{N}$ as preconditioner.

As a result, time complexity of each iteration of GMRES involves solving a delay-free Lyapunov equation with time complexity of $\mathcal{O}(n^3)$. More details about GMRES algorithm are provided in [38].

In order to complete the computation of $\rho(L,\tau)$, we should multiply z_1^{T} by z_2 in $\mathcal{O}(n^4)$. However, it can be done more efficiently if we first multiply $\mathcal{D}$ by z_2 by using (54) in $\mathcal{O}(n^3)$

and then multiply $\left[\mathrm{vec}(BB^{\mathrm{T}})^{\mathrm{T}} \quad 0_{1\times n^2}\right]$ by $\mathcal{D}\, z_2$ in $\mathcal{O}(n^2)$. Table I shows the number of GMRES iterations that is necessary to find the performance with relative residue 10^{-14}. Our observation is that the number of iterations is independent of size of the network, i.e., n. However, as the ratio of time-delay to time-delay margin increases, the number of iterations increase as well. Figure 4 depicts required time to evaluate performance of a network with different sizes and different amounts of time-delay. As one can expect, in the Figure 4 as size of the network increases, solving the delay Lyapunov equation becomes more time consuming. In addition, cost of performance computation increases with growth of the ratio $\frac{\tau}{\tau^*}$. This observation is compatible with relation between number of iterations and the ratio $\frac{\tau}{\tau^*}$ given in table I.

B. Gradient Computation

As it was mentioned earlier, another time consuming part in each iteration of optimization is gradient computation. Here we compute gradient of the objective function without computing the directional derivatives. As an example, derivative of the objective function with respect to weight of edge (i,j) is

$$\begin{aligned}\frac{\rho(L,\tau)}{\partial w_{ij}} &= \begin{bmatrix}\mathrm{vec}(BB^{\mathrm{T}})\\ 0_{n^2\times 1}\end{bmatrix}^{\mathrm{T}} \Big[\frac{\partial\,\mathcal{D}}{\partial w_{ij}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\\ &-\mathcal{D}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\Big(\frac{\partial\mathbf{M}}{\partial w_{ij}}\,\mathcal{D}+\mathbf{M}\frac{\partial\,\mathcal{D}}{\partial w_{ij}}\\ &+\frac{\partial\mathbf{N}}{\partial w_{ij}}\,\mathcal{N}+\mathbf{N}\frac{\partial\,\mathcal{N}}{\partial w_{ij}}\Big)(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\Big]\begin{bmatrix}-\mathrm{vec}\,M_n\\ 0_{n^2\times 1}\end{bmatrix}.\end{aligned} \tag{56}$$

Let us define δ_1, δ_2, δ_3, δ_4, and δ_5 as

$$\begin{aligned}\delta_1 &= \left[\mathrm{vec}(BB^{\mathrm{T}})^{\mathrm{T}} \quad 0_{1\times n^2}\right]\frac{\partial\,\mathcal{D}}{\partial w_{ij}}z_2,\\ \delta_2 &= z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\frac{\partial\mathbf{M}}{\partial w_{ij}}\,\mathcal{D}\,z_2,\\ \delta_3 &= z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\mathbf{M}\frac{\partial\,\mathcal{D}}{\partial w_{ij}}z_2,\\ \delta_4 &= z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\frac{\partial\mathbf{N}}{\partial w_{ij}}\,\mathcal{N}\,z_2,\\ \delta_5 &= z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}\mathbf{N}\frac{\partial\mathcal{N}}{\partial w_{ij}}z_2.\end{aligned} \tag{57}$$

Therefore we get the following equivalent relation for (56)

$$\frac{\partial\rho(L,\tau)}{\partial w_{ij}} = \delta_1-\delta_2-\delta_3-\delta_4-\delta_5. \tag{58}$$

Furthermore, $\frac{\partial L^{\mathrm{T}}}{\partial w_{ij}} = \left(\frac{\partial L}{\partial w_{ij}}\right)^{\mathrm{T}}$. Derivative of Laplacian matrix with respect to w_{ij} is

$$\left[\frac{\partial L}{\partial w_{ij}}\right]_{kl} = \begin{cases}1, & \text{if } k=l=i,\\ -1, & \text{if } k=i, l=j,\\ 0, & \text{otherwise,}\end{cases}$$

Hence, $\frac{\partial L}{\partial w_{ij}}$ can be represented compactly as $\mathbf{e}_i(\mathbf{e}_i-\mathbf{e}_j)^{\mathrm{T}}$. Moreover, $\frac{\partial\,\mathcal{N}}{\partial w_{ij}}$ and $\frac{\partial\,\mathcal{D}}{\partial w_{ij}}$ are given as

$$\frac{\partial\,\mathcal{N}}{\partial w_{ij}} = \frac{\partial V_1}{\partial w_{ij}}+\frac{\partial V_2}{\partial w_{ij}}, \tag{59}$$

$$\frac{\partial\,\mathcal{D}}{\partial w_{ij}} = -\frac{\partial V_1}{\partial w_{ij}}+\frac{\partial V_2}{\partial w_{ij}}, \tag{60}$$

where $\frac{\partial V_1}{\partial w_{ij}}$ and $\frac{\partial V_2}{\partial w_{ij}}$ are

$$\begin{aligned}\frac{\partial V_1}{\partial w_{ij}} &= \sum_{i=1}^{7}\beta_{2i-1}\tau^{2i-1}\sum_{j=0}^{2i-2}\mathbf{A}^j\frac{\partial\mathbf{A}}{\partial w_{ij}}\mathbf{A}^{2i-2-j},\\ \frac{\partial V_2}{\partial w_{ij}} &= \sum_{i=1}^{6}\beta_{2i}\tau^{2i}\sum_{j=0}^{2i-1}\mathbf{A}^j\frac{\partial\mathbf{A}}{\partial w_{ij}}\mathbf{A}^{2i-1-j},\end{aligned} \tag{61}$$

under the convention that $\mathbf{A}^0 = I_{2n^2}$. Consider two augmented vectors $u\in\mathbb{R}^{2n^2}$ and $\tilde{u}\in\mathbb{R}^{2n^2}$ as in (53). We have

$$\tilde{u}^{\mathrm{T}}\frac{\partial\mathbf{M}}{\partial w_{ij}}u, = [\tilde{U}_1^{\mathrm{T}}U_2]_{ji}-[\tilde{U}_1^{\mathrm{T}}U_2]_{ii}, \tag{62}$$

$$\tilde{u}^{\mathrm{T}}\frac{\partial\mathbf{N}}{\partial w_{ij}}u, = [U_1\tilde{U}_1^{\mathrm{T}}]_{ij}-[U_1\tilde{U}_1^{\mathrm{T}}]_{ii}, \tag{63}$$

$$\begin{aligned}\tilde{u}^{\mathrm{T}}\frac{\partial\mathbf{A}}{\partial w_{ij}}u &= [U_1\tilde{U}_2^{\mathrm{T}}]_{ii}-[U_1\tilde{U}_2^{\mathrm{T}}]_{ij}\\ &+[\tilde{U}_1^{\mathrm{T}}U_2]_{ji}-[\tilde{U}_1^{\mathrm{T}}U_2]_{ii}.\end{aligned} \tag{64}$$

At this point we have all the tools that we require to calculate $\frac{\partial\rho(L,\tau)}{\partial w_{ij}}$ as in (58). Let z_3^{T} equals $z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}$. From (57) we infer δ_2, δ_3, δ_4 and δ_5 include z_3^{T}. The procedure of finding z_3 is similar to what we have already explained for finding z_2 in performance calculation. To this end, the following equation should be solved for z_3 by GMRES algorithm

$$(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{\mathrm{T}}z_3 = z_1, \tag{65}$$

and to speed-up the convergence rate, we take $(\mathbf{M}+\mathbf{N})^{\mathrm{T}}$ as a right preconditioner. Finding product of $(\mathbf{M}^{\mathrm{T}}+\mathbf{N}^{\mathrm{T}})^{-1}$ and a vector can be done with $\mathcal{O}(n^3)$ arithmetic operations similar to product of $(\mathbf{M}+\mathbf{N})^{-1}$ and a vector that we explained for computing performance. Moreover, by the similar technique as in (54) matrix-vector multiplication for $(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{\mathrm{T}}$ needs $\mathcal{O}(n^3)$ operations. In other words for a vector u as in (53), cost of computing $\mathbf{M}^{\mathrm{T}}u$, $\mathbf{N}^{\mathrm{T}}u$ and $\mathbf{A}^{\mathrm{T}}u$ is $\mathcal{O}(n^3)$. After solving equation (65) and finding $z_3^{\mathrm{T}} = z_1^{\mathrm{T}}(\mathbf{M}\,\mathcal{D}+\mathbf{N}\,\mathcal{N})^{-1}$ computation of δ_1 and δ_3 take the form of (64). Similarly, by (59) and (61), computation of δ_5 takes the form of (64), as well. Moreover, computation of δ_2 and δ_4 end up with an equation similar to (62) and (63), respectively. Thus after computing z_2 and z_3 the rest of the steps requires $\mathcal{O}(n^3)$ operations.

C. Example

We run the optimization problem for randomly generated graphs with a number of nodes in range of 50 to 200. Figure 5 depicts the ratio of the result of the optimization (solution of of the problem (51)) to ρ_n^* in (41). For each data point in the figure, an Erdős-Renyi random graph is considered as the topology. As the density of the graphs increases, this ratio decreases, since we have more variables (edge weights) to adjust through the optimization process.

VII. Conclusion

The $\mathcal{H}_2$ performance analysis of a class of time-delay directed consensus networks is considered. When network Laplacian is normal, we obtain a closed-form expression for

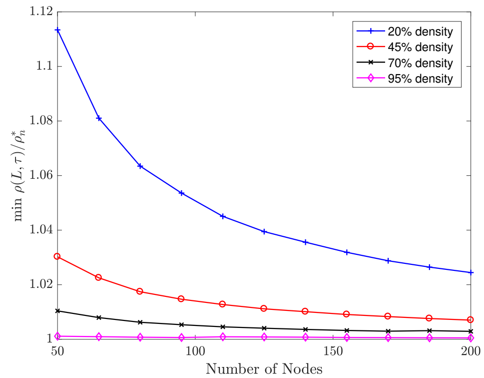

Fig. 5. Ratio of the optimal network performance provided by solving the optimization problem (51) to minimum of performance of networks with normal Laplacian matrix, given by (41). For each data point, an Erdős-Renyi random graph $\mathcal{G}(n,p)$ is generated, in which p is approximately the density. We observe that the best achievable performance is improved by increasing the density.

the $\mathcal{H}_2$-norm, which depends on Laplacian eigenvalues and time-delay. We show that network performance deteriorates when time-delay increase and its behavior is non-monotone in terms of feedback gains. We present a design algorithms that can tune feedback gains in a network (with a fixed topology) with several thousands state variable to achieve optimal performance.

References

[1] Q.-C. Zhong, *Robust control of time-delay systems*. Springer Science & Business Media, 2006.

[2] Y. Ghaedsharaf and N. Motee, "Performance improvement in noisy linear consensus networks with time-delay," *IEEE Transactions on automatic control*, vol. 64, 2019.

[3] C. Somarakis, Y. Ghaedsharaf, and N. Motee, "Time-delay origins of fundamental tradeoffs between risk of large fluctuations and network connectivity," *IEEE Transactions on automatic control*, 2019.

[4] ——, "Risk analysis of second-order models of vehicle platooning: Time-delay induced limitations and trade-offs," *IEEE Transactions on automatic control*, 2019.

[5] R. Sipahi, S.-I. Niculescu, C. T. Abdallah, W. Michiels, and K. Gu, "Stability and stabilization of systems with time delay," *IEEE Control Systems*, vol. 31, no. 1, pp. 38–65, 2011.

[6] N. Olgac and R. Sipahi, "An exact method for the stability analysis of time-delayed linear time-invariant (lti) systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 793–797, 2002.

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

[8] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[9] P.-A. Bliman and G. Ferrari-Trecate, "Average consensus problems in networks of agents with delayed communications," *Automatica*, vol. 44, no. 8, pp. 1985–1995, 2008.

[10] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on automatic control*, vol. 50, no. 1, pp. 121–127, 2005.

[11] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on automatic control*, vol. 50, no. 5, pp. 655–661, 2005.

[12] 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 Transactions on Automatic Control*, vol. 56, no. 4, pp. 814–825, 2011.

[13] G. F. Young, L. Scardovi, and N. E. Leonard, "Robustness of noisy consensus dynamics with directed communication," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6312–6317.

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

[15] B. Bamieh, M. Jovanovic, P. Mitra, and S. Patterson, "Effect of topological dimension on rigidity of vehicle formations: Fundamental limitations of local feedback," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*. IEEE, 2008, pp. 369–374.

[16] Y. Ghaedsharaf, M. Siami, C. Somarakis, and N. Motee, "Interplay between performance and communication delay in noisy linear consensus networks," in *Control Conference (ECC), 2016 European*. IEEE, 2016, pp. 1703–1708.

[17] I. M. Jaimoukha and E. M. Kasenally, "Krylov subspace methods for solving large lyapunov equations," *SIAM Journal on Numerical Analysis*, vol. 31, no. 1, pp. 227–251, 1994.

[18] E. Jarlebring and F. Poloni, "Iterative methods for the delay lyapunov equation with t-sylvester preconditioning," *arXiv preprint arXiv:1507.02100*, 2015.

[19] S. Dezfulian, Y. Ghaedsharaf, and N. Motee, "On performance of time-delay linear consensus networks with directed interconnection topologies," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 4177–4182.

[20] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[21] J. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[22] J. Hu and Y. Hong, "Leader-following coordination of multi-agent systems with coupling time delays," *Physica A: Statistical Mechanics and its Applications*, vol. 374, no. 2, pp. 853 – 863, 2007.

[23] R. Sipahi and A. Acar, "Stability analysis of three-agent consensus dynamics with fixed topology and three non-identical delays," in *ASME 2008 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2008, pp. 1483–1490.

[24] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robot. Automat. Mag.*, vol. 17, no. 3, pp. 56–65, 2010.

[25] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: The flying machine arena," *Mechatronics*, vol. 24, no. 1, pp. 41–54, 2014.

[26] M. Siami, S. Bolouki, B. Bamieh, and N. Motee, "Centrality measures in linear consensus networks with structured network uncertainties," *IEEE Transactions on Control of Network Systems*, 2017.

[27] M. Buslowicz, "Simple stability criterion for a class of delay differential systems," *International Journal of Systems Science*, vol. 18, no. 5, pp. 993–995, 1987.

[28] K. Gu, J. Chen, and V. L. Kharitonov, *Stability of time-delay systems*. Springer Science & Business Media, 2003.

[29] E. Plischke, "Transient effects of linear dynamical systems," Ph.D. dissertation, der Universitat Bremen, 2005.

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

[31] N. J. Higham, *Functions of matrices: theory and computation*. SIAM, 2008.

[32] T.-T. Lu and S.-H. Shiou, "Inverses of 2× 2 block matrices," *Computers & Mathematics with Applications*, vol. 43, no. 1-2, pp. 119–129, 2002.

[33] D. Hunt, G. Korniss, and B. K. Szymanski, "Network synchronization in a noisy environment with time delays: Fundamental limits and trade-offs," *Physical review letters*, vol. 105, no. 6, p. 068701, 2010.

[34] M. Siami and N. Motee, "New spectral bounds on h2-norm of linear dynamical networks," *Automatica*, vol. 80, pp. 305–312, 2017.

[35] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[36] N. J. Higham, "The scaling and squaring method for the matrix exponential revisited," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 4, pp. 1179–1193, 2005.

[37] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.

[38] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[39] Y. Saad, "A flexible inner-outer preconditioned gmres algorithm," *SIAM Journal on Scientific Computing*, vol. 14, no. 2, pp. 461–469, 1993.

[40] G. Golub, S. Nash, and C. Van Loan, "A hessenberg-schur method for the problem ax+ xb= c," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.