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 efficiently by another spectral function that can be used to enhance the network performance via growing, sparsification, and reweighting 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 state variables.

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

$\mathbb{R}^+(\mathbb{R}^+) \times \{0\}$ 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 $c = \Re(c) + \Im(c)j$, where $\Re(c)$ and $\Im(c)$ denote the real and imaginary parts of $c$, respectively. Also, $j$ is the imaginary unit, so $j^2 = -1$. $\mathcal{I}$ is a shorthand notation for the set $\{1, 2, \ldots, n\}$. Vector $e_i \in \mathbb{R}^n$ is a unit vector whose $i^{th}$ entry is equal to one and the rest of its entries are equal to zero. $1_n$ is the vector of all ones. For a matrix $M$, transpose, conjugate, conjugate transpose, and Moore-Penrose pseudoinverse are shown by $M^T$, $M^\dagger$, $M^{\dagger}$, 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 $M_n = I_n - \frac{1}{n}1_n1_n^T$. Kronecker product of matrix $M$ and matrix $N$ is denoted by $M \otimes N$. Vectorization of matrix $M$ is shown by vec$(M)$, which is a column vector. Trace of matrix $M$ is denoted by $\text{Tr}(M)$. Matrix $M \in \mathbb{C}^{n \times n}$ is unitary if $MM^H = M^HM = I_n$. Matrix $M \in \mathbb{C}^{n \times n}$ is normal if $MM^H = M^HM$.

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

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

where $w_{ij}$ is the weight of edge $(i,j) \in \mathcal{E}$. Let $k_{ij}$ to be the shorthand notation for $w((i,j))$ for all $(i,j) \in \mathcal{E} \times \mathcal{V}$. For any edge $(i,j) \in \mathcal{E}$, 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 $G$ is a sequence of distinct nodes of the graph which are connected to each other by a sequence of edges [20]. A graph $G$ has a globally reachable node if there exists a node such that there is a directed path from any other node of the graph $G$ to that node [21]. A node $j$ is in the neighborhood of node $i$, if $(i,j) \in \mathcal{E}$; therefore, set of neighbors of node $i$ is defined as $\mathcal{N}_i = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$. In-degree of the node $i$ is $d_i^\text{in} = \sum_{j=1}^n k_{ij}$, likewise, out-degree of node $i$ is $d_i^\text{out} = \sum_{j=1}^n k_{ji}$. A graph is balanced if and only if all of its nodes have the same in-degree and out-degree. Consider a vector $d = [d_1^\text{out}, d_2^\text{out}, \ldots, d_n^\text{out}]^T$, 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 $G$ is defined by

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

Laplacian matrix of graph $G$ is defined as $L = D - A$, where $D = \text{diag}(d)$. 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, $L_1 = 0$ and $L$ has at least one zero eigenvalue. Eigenvalues of Laplacian matrix are shown by $\lambda_i$’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, $L_n^2 L = 0$.

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 $G$, 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 $i \in \mathcal{V}$, 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

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

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 $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$. Vector of noise $\xi(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_n(t)]^T$ 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

$$\dot{x}(t) = -Lx(t - \tau) + \xi(t), \quad t \geq 0,$$

$$y(t) = Mx(t),$$

$$1\text{This assumption has been widely used by other researchers as it allows analytical derivations of formulas. For rendezvous in time, it is a common practice in robotics labs to use identical communication modules for all agents, which results in a uniform communication time-delay. Moreover, in other related applications such as heading alignments, rendezvous in space, and velocity control of agents using a Motion Capture (MoCap) system to observe their spatial locations in indoor labs [23]–[25] all agents experience an identical time-delay to access data through MoCap system.}$$
with initial condition
\[ x(t) = \begin{cases} 0 & \text{if } t \in [-\tau, 0) \\ x_0 & \text{if } t = 0. \end{cases} \]  
(3)

Matrix \( L \) is the Laplacian matrix of the underlying communication graph \( G \) 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 \( G \) has a globally reachable node.

By the above assumption, \( 1_n \) 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 \( H_2 \)-norm of the network as the performance measure, which is defined as

\[ \rho(L, \tau) = \mathbb{E} \left[ \int_{t=0}^{\infty} \text{Tr} \left( (y(t))^T y(t) \right) dt \right], \]  
(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 \( H_2 \)-norm increases. In this paper, we seek a closed-form expression for the \( H_2 \)-norm of the network (2) when the Laplacian matrix of the communication graph \( G \) 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 \( H_2 \)-norm of network (2) and its gradient in order to optimize the performance by allocating optimal edge weights to the graph \( G \).

**Remark 1.** Let \( \nu \) span the left null space of Laplacian matrix. It can be shown that in the absence of noise, the value of consensus is \( \frac{\nu x_0}{\nu^T 1_n} \). It is worth mentioning that in section VI, we use the notion of average-consensus. Similarly 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 \( \frac{1_n \nu x_0}{\nu^T 1_n} = z_1(0)+z_2(0)+\cdots+z_n(0) \).

**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 \( H_2 \)-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

\[ \dot{x}(t) = -\hat{L} \hat{x}(t - \tau) + \left( I_n - \frac{1_n \nu^T}{\nu^T 1_n} \right) \xi(t), \quad t \geq 0, \]  
(5)

\[ y(t) = M_n \hat{x}(t), \]

with initial condition

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

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

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

\[ y = M_n \hat{x}(t) = M_n x(t). \]  
(7)

Transfer matrix of (2) and (5) are

\[ G_L(s) = M_n (sI + e^{-s \tau} L)^{-1}, \]  
(8)

\[ G_{\hat{L}}(s) = M_n (sI + e^{-s \tau} \hat{L})^{-1} (I_n - \frac{1_n \nu^T}{\nu^T 1_n}), \]

respectively. The Jordan normal form of matrices \( L \) and \( \hat{L} \) are given by

\[ L = P \text{diag}([0, J_2, \ldots, J_p]) P^{-1} = P J_L P^{-1}, \]

\[ \hat{L} = P \text{diag}([\alpha, J_2, \ldots, J_p]) P^{-1} = P J_{\hat{L}} P^{-1}, \]  
(9)

where \( \text{diag}([0, J_2, \ldots, J_p]) \) is a block diagonal matrix such that \( 0, J_2, \ldots, J_p \) are its main diagonal blocks, moreover, \( \text{diag}([\alpha, J_2, \ldots, J_p]) \) is similarly defined.

Also, it can be shown that

\[ (I_n - \frac{1_n \nu^T}{\nu^T 1_n}) = P \text{diag}([0,1, \ldots,1]) P^{-1} = P \Lambda_T P^{-1}, \]

(10)

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

\[ G_L(s) = M_n P (sI_n + e^{-s \tau} J_L)^{-1} P^{-1}, \]

\[ G_{\hat{L}}(s) = M_n P (sI_n + e^{-s \tau} J_{\hat{L}})^{-1} \Lambda_T P^{-1}. \]  
(11)

Note that

\[ (sI_n + e^{-s \tau} J_L)^{-1} = \text{diag}([s^{-1}, J_2^{-1}, \ldots, J_p^{-1}]), \]

\[ (sI_n + e^{-s \tau} J_{\hat{L}})^{-1} = \text{diag}([s + e^{-s \tau} \alpha d^{-1}, J_2^{-1}, \ldots, J_p^{-1}]), \]  
(12)

where \( J_i^{-1} = (sI_{n(i)} + e^{-s \tau} J_i)^{-1} \) 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

\[ G_L(s) = M_n P \text{diag}([0, J_2^{-1}, \ldots, J_p^{-1}]) P^{-1}, \]

\[ G_{\hat{L}}(s) = M_n P \text{diag}([0, J_2^{-1}, \ldots, J_p^{-1}]) P^{-1}. \]
Proposition 1. The input matrix of the modified system (5) is real-valued.

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

\[
\nu^T L = \nu_1^T L + \nu_2^T L j = 0.
\]

(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 \( \nu_1 \) and \( \nu_2 \). It is a contradiction since we have assumed that rank of \( L \) is \( n - 1 \). As a result, \( \nu \) 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 \( i \in \mathbb{I} \), \( \Re{\lambda_i} > 0 \), and \( \tau \) is less than time-delay margin \( \tau^* \), i.e., \( \tau \in [0, \tau^*) \) [27], where the time-delay margin is

\[
\tau^* = \min_i \left( \frac{1}{|\alpha_i|} \arcsin \left( \frac{\Re{\lambda_i}}{|\alpha_i|} \right) \right).
\]

(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 \( \alpha_d \). 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 \( G_L \) 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

\[
\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr} \left( G_L(\omega)^H G_L(\omega) \right) d\omega.
\]

(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 \( (sI_n + \tilde{L} e^{-\sigma t})^{-1} \), i.e., \( L[K] = (sI_n + \tilde{L} e^{-\sigma t})^{-1} \), and satisfies the equation

\[
\frac{d}{dt} K(t) = -\tilde{L} K(t - \tau),
\]

(16)

for \( t \geq 0 \), and the initial conditions

\[
K(0) = I_n \text{ and } K(t) = 0 \quad \text{for} \quad t < 0.
\]

(17)

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

\[
\frac{d}{dt} K(t) = -K(t - \tau) \tilde{L},
\]

(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 \( \mathcal{H}_2 \)-Lyapunov matrix of the system (5) is defined as

\[
U(t) := \int_{s=0}^{\infty} K^T(s) M_n^T M_n K(s + t) d\omega.
\]

(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

\[
U(-t) = U^T(t), \quad \text{for} \quad t \in [-\tau, \tau]
\]

\[
U^T(\tau) \tilde{L} + \tilde{L}^T U(\tau) = M_n,
\]

\[
\dot{U}(t) = -U(t - \tau) \tilde{L}, \quad \text{for} \quad t \in [0, \tau].
\]

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

\[
\text{Tr} \left( \left( I_n - \frac{1}{\nu^T L_1} \right)^T \left( I_n - \frac{1}{\nu^T L_1} \right) \right).
\]

(20)

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

\[
B := M + Ne^{A\tau}.
\]

(21)

where

\[
A := \begin{bmatrix} 0 & -\tilde{L} \otimes I_n \\ I_n \otimes \tilde{L}^T & 0 \end{bmatrix},
\]

\[
M := \begin{bmatrix} 0 & -\tilde{L} \otimes I_n \\ I_n \otimes \tilde{L}^T & 0 \end{bmatrix},
\]

\[
N := \begin{bmatrix} -I_n \otimes \tilde{L}^T & 0 \\ 0 & -I_n \otimes \tilde{L}^T \end{bmatrix}.
\]

(22)

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

\[
\text{vec} U(0) = \begin{bmatrix} I_{n^2} & 0 \end{bmatrix} B^{-1} \begin{bmatrix} -\text{vec} M_n \\ 0_{n^2 \times 1} \end{bmatrix}.
\]

(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. $H_2$-norm of Time-delay Network with Normal Laplacian

Normal modified Laplacian matrix $\tilde{L}$ can be written as $\tilde{L} = Q\Lambda_{\tilde{L}}Q^H$, 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 $H_2$-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 $G$ with globally reachable node and normal Laplacian matrix is balanced.

Theorem 2. For system (2) with corresponding normal Laplacian matrix, the $H_2$-norm squared is represented in terms of time-delay and Laplacian matrix eigenvalues as

$$\frac{1}{2} \sum_{i=1}^{n} \frac{\cos (\tau_i |\lambda_i|)}{\Re\{\lambda_i\} - |\lambda_i| \sin (\tau_i |\lambda_i|)}.$$  \hspace{1cm} (24)

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

$$\tilde{L}^H \otimes I_n = (Q \otimes Q)(\tilde{A}_L \otimes I_n)(Q^H \otimes Q^T),$$

$$I_n \otimes \tilde{L}^T = (Q \otimes Q)(I_n \otimes \Lambda_{L})(Q^H \otimes Q^T).$$  \hspace{1cm} (25)

Set matrix $\Omega$ and matrix $\Xi$ to be as follows

$$\Omega = \begin{bmatrix} Q \otimes Q & 0_{n^2 \times n^2} \\ 0_{n^2 \times n^2} & Q \otimes Q \end{bmatrix}, \quad \Xi = \begin{bmatrix} I_n \otimes \Lambda_L & -\tilde{A}_L \otimes I_n \\ I_n \otimes \Lambda_L & 0_{n^2 \times n^2} \end{bmatrix}.$$  \hspace{1cm} (26)

Therefore, we can rewrite matrix $A$ as

$$A = \Omega \Xi Q^H.$$  \hspace{1cm} (27)

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

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

It follows that

$$M = \Omega \Lambda_M Q^H, \quad N = \Omega \Lambda_N Q^H,$$  \hspace{1cm} (28)

where $\Lambda_M$ and $\Lambda_N$ are

$$\Lambda_M = \begin{bmatrix} 0_{n^2 \times n^2} & -\Lambda_L^H \otimes I_n \\ I_{n^2} & 0_{n^2 \times n^2} \end{bmatrix}, \quad \Lambda_N = \begin{bmatrix} -I \otimes \Lambda_L & 0 \\ 0_{n^2 \times n^2} & -I_{n^2} \end{bmatrix}.$$  \hspace{1cm} (29)

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

$$e^{A\tau} = \Omega e^{\Xi \tau} Q^H.$$  \hspace{1cm} (30)

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

$$(M + Ne^{A\tau})^{-1} = \Omega \left(\Lambda_M + \Lambda_N e^{\Xi \tau}\right)^{-1} Q^H.$$  \hspace{1cm} (31)

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

$$\text{vec}(U(0)) = \begin{bmatrix} I_{n^2} & 0_{n^2 \times n^2} \end{bmatrix} \Omega (\Lambda_M + \Lambda_N e^{\Xi \tau})^{-1} Q^H \begin{bmatrix} -\text{vec}(M) \\ 0 \end{bmatrix}.$$  \hspace{1cm} (32)

Let us compute $e^{\Xi \tau}$ 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

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

then $f(M)$ is as follows

$$f(M) = Pf(\Sigma)P^{-1} = P \text{diag}(\{f(\sigma_1), \ldots, f(\sigma_n)\}) P^{-1}.$$  \hspace{1cm} (33)

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

$$e^{\Omega \tau} = I_{2n^2} \otimes \Omega \tau + \Omega^2 \tau^2 \otimes \frac{\Omega^3 \tau^3}{3!} + \ldots.$$  \hspace{1cm} (34)

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

$$\Omega^{2k} = (-1)^k \begin{bmatrix} \Lambda_L^k \otimes \Lambda_L^k & 0_{n^2 \times n^2} \\ 0_{n^2 \times n^2} & \Lambda_L^k \otimes \Lambda_L^k \end{bmatrix},$$

$$\Omega^{2k-1} = (-1)^k \begin{bmatrix} 0_{n^2 \times n^2} & \Lambda_L^k \otimes \Lambda_L^{k-1} \\ -\Lambda_L^{k-1} \otimes \Lambda_L^k & 0_{n^2 \times n^2} \end{bmatrix}.$$  \hspace{1cm} (35)

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

$$(e^{\Omega \tau})_{11} = \cos \left(\frac{\Lambda_L}{2} \tau\right),$$

$$(e^{\Omega \tau})_{12} = -\frac{\Lambda_L}{2} \sin \left(\frac{\Lambda_L}{2} \tau\right) \Lambda_L^{-\frac{1}{2}},$$

$$(e^{\Omega \tau})_{21} = \frac{\Lambda_L}{2} \sin \left(\frac{\Lambda_L}{2} \tau\right) \Lambda_L^{-\frac{1}{2}},$$

$$(e^{\Omega \tau})_{22} = \cos \left(\frac{\Lambda_L}{2} \tau\right).$$  \hspace{1cm} (36)

It follows that

$$\Lambda_M + \Lambda_N e^{\Xi \tau} = \begin{bmatrix} (\Lambda_M + \Lambda_N e^{\Xi \tau})_{11} & (\Lambda_M + \Lambda_N e^{\Xi \tau})_{12} \\ (\Lambda_M + \Lambda_N e^{\Xi \tau})_{21} & (\Lambda_M + \Lambda_N e^{\Xi \tau})_{22} \end{bmatrix}.$$  \hspace{1cm} (37)

Where $(\Lambda_M + \Lambda_N e^{\Xi \tau})_{ij}$ for $i, j \in \{1, 2\}$ are given by

$$(\Lambda_M + \Lambda_N e^{\Xi \tau})_{11} = -(I_n \otimes \Lambda_L)(e^{\Xi \tau})_{11},$$

$$(\Lambda_M + \Lambda_N e^{\Xi \tau})_{12} = -(I_n^H \otimes I_n) - (I_n \otimes \Lambda_L)(e^{\Xi \tau})_{12},$$

$$(\Lambda_M + \Lambda_N e^{\Xi \tau})_{21} = I_{n^2} - (e^{\Xi \tau})_{21},$$

$$(\Lambda_M + \Lambda_N e^{\Xi \tau})_{22} = -(e^{\Xi \tau})_{22}.$$  \hspace{1cm} (38)

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

$$\left(I_n - \frac{1}{\nu^T I_n}\right) = \left(I_n - \frac{1}{\nu^T I_n}\right)^H.$$  \hspace{1cm} (39)

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

$$\text{vec}(QA_TQ^H U(0)) = \text{vec}(I_n - \frac{1}{\nu^T I_n} I_n U(0)).$$  \hspace{1cm} (40)

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

$$\text{Tr}(M^H \tilde{M}) = \text{vec}(M)^H \text{vec}(\tilde{M}).$$

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

$$\text{vec}(QA_TQ^H U(0)) = \text{vec}(U(0)).$$  \hspace{1cm} (41)
By premultiplying equation (29) by vec$(QA_T Q^H)^T$ we obtain
\[
[vec(QA_T Q^H)^T 0] \Omega (A_M + \Lambda N e^{\Omega r})^{-1} Q^H \left[ -vec M_n \right] = [vec(QA_T Q^H)^T (Q \otimes \hat{Q}) 0] (A_M + \Lambda N e^{\Omega r})^{-1}
\]
\[
\left[-(Q^H \otimes Q^T) vec M_n^T \right] = -vec(A_T)^T (Q^H \otimes Q^T) (Q \otimes \hat{Q})
\]
\[
((A_M + \Lambda N e^{\Omega r})^{-1})_{11} (Q^H \otimes Q^T) vec(QA_M e^{\Omega r} M^T)
\]
\[
= -vec(A_T)^T ((A_M + \Lambda N e^{\Omega r})^{-1})_{11} vec(M_n).
\]

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]
\[(P^{-1})_{11} = (P_{11} - P_{12} P_{22}^{-1} P_{21})^{-1}.
\]

Hence, by substituting from (33) into (37) we get
\[
((A_M + \Lambda N e^{\Omega r})^{-1})_{11} = -\left( (I_n \otimes \Lambda L) \cos \left( (\hat{A}_L \otimes A_L)^{1/2} \tau \right) + [(\hat{A}_L \otimes I_n) - (I_n \otimes \Lambda L) \hat{A}_L^{1/2} \sin \left( (\hat{A}_L \otimes A_L)^{1/2} \tau \right) - \hat{A}_L^{1/2}] \right)^{-1} \left( [I_n, z - \Lambda L^{1/2} \sin \left( (\hat{A}_L \otimes A_L)^{1/2} \tau \right) \Lambda L^{1/2}] \right)^{-1}
\]

Since $\Lambda L$ and $\hat{A}_L$ are diagonal matrices, their kronecker product, i.e., $(\hat{A}_L \otimes A_L)$ is a diagonal matrix as well, hence $\sin((\hat{A}_L \otimes A_L)^{1/2} \tau)$ and $\cos((\hat{A}_L \otimes A_L)^{1/2} \tau)$ 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 $((A_M + \Lambda N e^{\Omega r})^{-1})_{11}$ and then after multiplying it by vec$(A_M)$ from right and by $(-vec(A_T)^T)$ from left we get the equation (24).

For delay-free case, equation (24) is simplified to
\[
\frac{1}{2} \sum_{i=2}^{n} \frac{1}{\Re\{\lambda_i\}},
\]
which 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
\[
\frac{1}{2} \sum_{i=2}^{n} \cos(\tau\lambda_i) \lambda_i (1 - \sin(\tau\lambda_i)),
\]
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 ($\tau$) we get the following
\[
\frac{1}{2} \sum_{i=2}^{n} \left| \lambda_i \right|^2 \left( \Re(\lambda_i) - |\lambda_i| \sin(\tau(\left| \lambda_i \right|)) \right)^2.
\]

Our system is exponentially stable, therefore the denominator of equation (38) is positive and for every $i \in \{2, ..., n\}$ we have
\[
|\lambda_i| \sin(\tau(\left| \lambda_i \right|)) < \Re(\lambda_i).
\]

Multiplying both sides of the (39) by $(-\Re(\lambda_i))$ we get
\[
-|\lambda_i| \Re(\lambda_i) \sin(\tau(\left| \lambda_i \right|)) > -\Re(\lambda_i)^2.
\]

Add $|\lambda_i|^2$ to the both sides of the equation (40)
\[
|\lambda_i|^2 - \sin(\tau(\left| \lambda_i \right|))|\lambda_i| \Re(\lambda_i) > |\lambda_i|^2 - \Re(\lambda_i)^2,
\]

since $|\lambda_i|^2 - (\Re(\lambda_i))^2 \geq 0$ we obtain
\[
|\lambda_i|^2 - \sin(\tau(\left| \lambda_i \right|))|\lambda_i| \Re(\lambda_i) > 0.
\]

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
\[
\rho_n^* \approx 1.5319(n-1)\tau.
\]

**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.,
\[
\frac{1}{2} \sum_{i=2}^{n} \cos(\tau|\lambda_i|) \left( \Re|\lambda_i| - |\lambda_i| \sin(\tau|\lambda_i|) \right).
\]

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-consensus 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, $L_{i}$ is Normal, then the modified Laplacian matrix, $\tilde{L}$, is also normal.

**Directed star graph $S_{n}$:** 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
\[
L_{S_{n}} = \begin{bmatrix}
0 & 0 & \ldots & 0 & 0 \\
-1 & 1 & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & 0 & 0 \\
-1 & 0 & \ldots & 1 & 0 \\
-1 & 0 & \ldots & 0 & 1
\end{bmatrix}.
\]
Eigenvalue of \( L_{S_n} \) 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 \( L_{S_n} \) is not a normal matrix. But if we choose \( \alpha_d = 1 \), then the modified Laplacian \( \tilde{L}_{S_n} \) is normal. From (24) we obtain the \( H_2 \)-norm squared
\[
\rho(\tilde{L}_{S_n}, \tau) = \frac{(n-1)}{2} \tan \left( \frac{\pi}{4} + \frac{\tau}{2} \right).
\]

Undirected star graph \( \tilde{S}_n \): 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
\[
\tilde{L}_{S_n} = \begin{bmatrix}
\frac{n-1}{2} & -\frac{1}{2} & \cdots & -\frac{1}{2} & -\frac{1}{2} \\
-\frac{1}{2} & \frac{n}{2} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
-\frac{1}{2} & 0 & \cdots & \frac{n}{2} & 0 \\
-\frac{1}{2} & 0 & \cdots & 0 & \frac{n}{2}
\end{bmatrix}.
\]

Eigenvalues of \( \tilde{L}_{S_n} \) are \( \{0, \frac{1}{2}, \ldots, \frac{n}{2} \} \). As a result, from (14), we obtain the corresponding time-delay margin \( \tau^*(n) = \frac{\pi}{n} \), which is a decreasing function of number of nodes. Since \( L_{K_n} \) is symmetric and normal, \( H_2 \)-norm squared is
\[
\rho(L_{K_n}, \tau) = \frac{(n-1)}{2} \tan \left( \frac{\pi}{4} + \frac{\tau}{2n-1} \right).
\]

Directed cycle graphs \( C_n \): 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
\[
L_{C_n} = I_n - \begin{bmatrix}
\mathbf{e}_2^T \\
\vdots \\
\mathbf{e}_n^T \end{bmatrix} = \begin{bmatrix}
1 & -1 & 0 & \cdots & 0 \\
0 & 1 & -1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & -1 \\
-1 & 0 & \cdots & 0 & 1
\end{bmatrix}.
\]

Eigenvalues of \( L_{C_n} \) are
\[
\lambda_{i+1} = 1 + e^{\pi(1 - \frac{2i}{n})} \quad \text{for } i \in \{0, 1, \ldots, n-1\}.
\]

Therefore, we have
\[
\Re(\lambda_{C_n,i}) = 2 \sin^2 \left( \frac{\pi}{n} i \right), \quad |\lambda_{C_n,i}| = 2 \sin \left( \frac{\pi}{n} i \right). \tag{42}
\]

As a result, from (14), we obtain the corresponding time-delay margin \( \tau^*(n) = \frac{\pi}{2n \sin \pi n} \), which is a decreasing function of \( n \). Hence, \( \lim_{n \to \infty} \tau^*(n) = \frac{1}{2} \). It can be shown that \( L_{C_n} \) is a normal matrix. Therefore, by (24) we get
\[
\rho(L_{C_n}, \tau) = \frac{1}{4} \sum_{i=1}^{n-1} \cos(2\tau \sin(\frac{\pi}{n} i)) \left( \sin(\frac{\pi}{n} i) - \sin(2\tau \sin(\frac{\pi}{n} i)) \right).
\]

In order to find an approximation of the \( H_2 \)-norm squared for \( 0 < x < \pi \), we define function \( f(x) \) as
\[
f(x) := \frac{\cos(2\tau \sin(x))}{\sin(x)(\sin(x) - \sin(2\tau \sin(x)))}.
\]

Now set
\[
f_1(x) = \lim_{x \to \pi^+} f(x) = \frac{1}{1 - 2\tau \pi^2}, \quad f_2(x) = \lim_{x \to \pi^-} f(x) = \frac{1}{1 - 2\tau (\pi - x)^2}.
\]

From (43) and (44), we obtain
\[
f_{\text{approx}}(x) = \frac{1}{1 - 2\tau \left( \frac{1}{x^2} + \frac{1}{(x-\pi)^2} - \frac{8}{\pi^2} \right) + \tan(\frac{\pi}{4} + \tau)}, \tag{45}
\]

where \( f_{\text{approx}}(x) \) is the approximation of function \( f(x) \). Set \( x = \frac{\pi}{n} \) in equation (45). We get the following approximation for the \( H_2 \)-norm squared of the cycle graph
\[
\frac{n^2}{2\pi^2(1 - 2\tau)} \sum_{i=1}^{n-1} \frac{1}{i^2} + \frac{n-1}{4} \left( \tan(\frac{\pi}{4} + \tau) - \frac{8}{\pi^2(1 - 2\tau)} \right). \tag{46}
\]

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

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 $\tilde{C}_n$: 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

$$L_{\tilde{C}_n} = \frac{L_{C_n} + L_{C_n}^T}{2},$$

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

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

Directed path graph $P_n$: 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

$$L_{P_n} = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & -1 \\ 0 & 0 & \ldots & 0 & 0 \end{bmatrix}.$$

Eigenvalues of $L_{P_n}$ are $\{0, 1, \ldots, 1\}$. As a result, from (14), we obtain the corresponding time-delay margin $\tau^* = \frac{\pi}{4}$. It can be shown that $L_{P_n}$ is not normal, therefore, we cannot apply formula (24) to compute $\mathcal{H}_2$-norm squared.

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

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

Eigenvalues of $L_{\tilde{P}_n}$ 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, $\tau^*(n) = \frac{\pi}{4 \cos(\frac{\pi}{2n})}$, which is a function of number of nodes. Since $L_{\tilde{P}_n}$ is symmetric, the $\mathcal{H}_2$-norm squared is

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

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 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 counterparts as we can see in Figure 3. Moreover, directed star graph out-performs undirected star graph like 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 $\tau$ 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 resulting 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

$$
\begin{array}{c}
vec U(0) = [I_{n^2} \ 0_{n^2}] B^{-1} \begin{bmatrix}
- vec M_n \\
n_0 \times 1
\end{bmatrix} \\
= [I_{n^2} \ 0_{n^2}] (M + N e^{A \tau})^{-1} \begin{bmatrix}
- vec M_n \\
n_0 \times 1
\end{bmatrix}.
\end{array}
$$

Therefore, from (20) we obtain

$$
\begin{array}{c}
\rho(L, \tau) = \text{Tr} \left( (I_n - \frac{1}{\nu^T} I_n) (I_n - \frac{1}{\nu^T} I_n)^T U(0) \right) \\
= \left( \text{vec} \left( (I_n - \frac{1}{\nu^T} I_n) (I_n - \frac{1}{\nu^T} I_n)^T \right) \right)^T \text{vec} U(0).
\end{array}
$$

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 $e^{A \tau}$. 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 $N_{p,q}(M)$ and $D_{p,q}(M)$ as

$$
N_{p,q}(M) = \sum_{j=0}^{p} \frac{(p + q - j)!}{(p + q - j)! j!} M^j,
$$

$$
D_{p,q}(M) = \sum_{j=0}^{q} \frac{(p + q - j)!}{(p + q)!} M^j.
$$

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

$$
\mathcal{R}_{p,q}(M) := D_{p,q}(M)^{-1} N_{p,q}(M).
$$

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., $\mathcal{R}_{p,q}(e^{A\tau}) \approx e^{A\tau}$. Let $D$ and $N$ denote the shorthand notations for $D_{p,q}(A\tau)$ and $N_{p,q}(A\tau)$, respectively. Hence, the objective function (50) can be written as

$$
\rho(L, \tau) = \begin{bmatrix}
\text{vec}(B B^T)^T \\
n_0 \times 1
\end{bmatrix}^T D (M D + NN)^{-1} \begin{bmatrix}
- vec M_n \\
n_0 \times 1
\end{bmatrix}.
$$

Now, if

$$
\begin{array}{c}
z_1^T = \begin{bmatrix}
\text{vec}(B B^T)^T \\
n_0 \times 1
\end{bmatrix} D, \\
z_2 = (M D + NN)^{-1} \begin{bmatrix}
- vec M_n \\
n_0 \times 1
\end{bmatrix},
\end{array}
$$

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

Let us remark that $\mathcal{R}_{i,i}$, where $i = \text{max}(p, q)$, 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 $\mathcal{R}_{p,q}(e^{A\tau})$, we should construct matrix $D$ and $N$. Define $V_1$ and $V_2$ as

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

where $\beta_i$’s are the coefficients of [13/13] Padé approximants that are as follows.
Define an arbitrary augmented vector $u \in \mathbb{R}^{2n^2}$ as

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

(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 vec($U_1$) = $u_1$ and vec($U_2$) = $u_2$. By (22) for $n \geq 1$ we obtain

$$M_u = \begin{bmatrix} -\text{vec}(U_2 \tilde{L}) \\ \text{vec}(U_1 \tilde{L}) \end{bmatrix}, \quad N_u = \begin{bmatrix} \text{vec}(\tilde{L}^T U_1) \\ \text{vec}(\tilde{L}^T U_2) \end{bmatrix},$$

(54)

$$A^{2n}u = (-1)^n \begin{bmatrix} \text{vec}((\tilde{L}_n)^T U_1 \tilde{L}_n) \\ \text{vec}((\tilde{L}_n)^T U_2 \tilde{L}_n) \end{bmatrix},$$

$$A^{2n-1}u = (-1)^n \begin{bmatrix} \text{vec}((\tilde{L}_n-1)^T U_2 \tilde{L}_n) \\ -\text{vec}((\tilde{L}_n-1)^T U_1 \tilde{L}_n-1) \end{bmatrix},$$

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

Now we have all the tools that we need in order to compute $\rho(L, \tau) = z_1^T z_2$. We start by computing $z_2$. Recall that $z_2 = (M \mathcal{D} + N N)^{-1} \begin{bmatrix} -\text{vec} M_n \\ 0_n \times 1 \end{bmatrix}$. To avoid computing matrix inverse for finding $z_2$ with time complexity $O(n^3)$, one can think of finding $z_2$ as solving

$$(M \mathcal{D} + N N)z_2 = \begin{bmatrix} -\text{vec} M_n \\ 0_n \times 1 \end{bmatrix}$$

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 $M \mathcal{D} + N N$ and a vector $u$ is $O(n^3)$. However, if matrix $M \mathcal{D} + N N$ 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 $\tau = 0$, $M \mathcal{D} + N N$ equals $\beta_0(M + N)$, one can employ $M + N$ as a left preconditioner. When time-delay is zero, $M + N$ is a perfect preconditioner since the condition number of $(M + N)^{-1}(M \mathcal{D} + N N)$ is one. Let $T = (I_n \otimes (-\tilde{L})^T) + (-I_n^T \otimes I_n)$, then inverse of $M + N$ is

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

Consider vector $u$ as (53), then we have

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

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 $O(n^3)$ by solving delay-free Lyapunov equation. As an example, let vec($X$) = $T^{-1}u_1$, then we have

$$u_1 = T \text{vec}(X) = \left( (I_n \otimes (-\tilde{L})^T) + (-\tilde{L}^T \otimes I_n) \right) \text{vec}(X) = \text{vec}(-\tilde{L}^T X) + \text{vec}(-X \tilde{L}),$$

which is equivalent to

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

(55)

where the above equation is delay-free Lyapunov equation. Hence, time-complexity of calculating $\mathcal{P}^{-1}u$ is $O(n^3)$. 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 $M + N$ as preconditioner. As a result, time complexity of each iteration of GMRES involves solving a delay-free Lyapunov equation with time complexity of $O(n^3)$. 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 $O(n^4)$. However, it can be done more efficiently if we first multiply $\mathcal{D}$ by $z_2$ by using (54) in $O(n^3)$.
and then multiply $[\text{vec}(BB^T)]^T 0_{1 \times n^2}$ by $D z_2$ in $O(n^2)$. 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 $\frac{\tau}{T}$. This observation is compatible with relation between number of iterations and the ratio $\frac{\tau}{T}$ 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 with respect to weight of edge $(i,j)$ as

$$\rho(L, \tau) = \left[ \text{vec}(BB^T) \right]^T \left[ \frac{\partial D}{\partial w_{ij}} (M D + N N)^{-1} \right. - \left. \frac{\partial N}{\partial w_{ij}} (M D + N N)^{-1} \right] \left[ \text{vec}(M_n) \right]_0 \times 1 \times 1.$$

Let us define $\delta_1, \delta_2, \delta_3, \delta_4$, and $\delta_5$ as

$$\begin{align*}
\delta_1 &= \left[ \text{vec}(BB^T) \right]^T \frac{\partial D}{\partial w_{ij}} z_2, \\
\delta_2 &= z_1^T (M D + N N)^{-1} \frac{\partial M}{\partial w_{ij}} D z_2, \\
\delta_3 &= z_1^T (M D + N N)^{-1} M \frac{\partial D}{\partial w_{ij}} z_2, \\
\delta_4 &= z_1^T (M D + N N)^{-1} \frac{\partial N}{\partial w_{ij}} N z_2, \\
\delta_5 &= z_1^T (M D + N N)^{-1} N \frac{\partial N}{\partial w_{ij}} z_2.
\end{align*}$$

Therefore we get the following equivalent relation for (56)

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

Furthermore, $\frac{\partial L}{\partial w_{ij}} = \left( \frac{\partial L}{\partial w_{ij}} \right)^T$, 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, $\frac{\partial N}{\partial w_{ij}}$ can be represented compactly as $e_i(e_i - e_j)^T$. Moreover, $\frac{\partial N}{\partial w_{ij}}$ and $\frac{\partial D}{\partial w_{ij}}$ are given as

$$\begin{align*}
\frac{\partial N}{\partial w_{ij}} &= \frac{\partial V_1}{\partial w_{ij}} + \frac{\partial V_2}{\partial w_{ij}}, \\
\frac{\partial D}{\partial w_{ij}} &= - \frac{\partial V_1}{\partial w_{ij}} + \frac{\partial V_2}{\partial w_{ij}}, \quad (59, 60)
\end{align*}$$

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

$$\begin{align*}
\frac{\partial V_1}{\partial w_{ij}} &= \sum_{i=1}^{7} \beta_{2i-1}^j \frac{\partial A}{\partial w_{ij}} A_{2i-2-j}, \\
\frac{\partial V_2}{\partial w_{ij}} &= \sum_{i=1}^{6} \beta_{2i} \frac{\partial A}{\partial w_{ij}} A_{2i-1-j} \quad (61)
\end{align*}$$

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

$$\begin{align*}
\tilde{u}^T \frac{\partial M}{\partial w_{ij}} u &= [\tilde{U}_1^T U_2]_{ij} - [\tilde{U}_1^T U_2]_{ii}, \\
\tilde{u}^T \frac{\partial N}{\partial w_{ij}} u &= [U_1^T U_1]_{ij} - [U_1^T U_1]_{ii}, \\
\tilde{u}^T \frac{\partial A}{\partial w_{ij}} u &= [U_1^T U_2]_{ii} - [U_1^T U_2]_{ij} + [\tilde{U}_1 U_2]_{ij} - [\tilde{U}_1 U_2]_{ii} \quad (62, 63, 64)
\end{align*}$$

At this point we have all the tools that we need to calculate $\frac{\partial \rho(L, \tau)}{\partial w_{ij}}$ in (58). Let $z_1^T$ equals $z_1^T (M D + N N)^{-1}$. From (57) we infer $\delta_2, \delta_3, \delta_4$, and $\delta_5$ include $z_1^T$. 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

$$\left( M D + N N \right) z_3 = \frac{\partial L}{\partial w_{ij}} \quad (65)$$

and to speed-up the convergence rate, we take $(M + N)^T$ as a right preconditioner. Finding product of $(M + N)^T$ and a vector can be done with $O(n^3)$ arithmetic operations similar to product of $(M + N)^{-1}$ and a vector that we explained for computing performance. Moreover, by the similar technique as in (54) matrix-vector multiplication for $(M D + N N)^T$ needs $O(n^3)$ operations. In other words for a vector $u$ as in (53), cost of computing $M^T u$, $N^T u$ and $A^T u$ is $O(n^3)$. After solving equation (65) and finding $z_3^T = \frac{\partial L}{\partial w_{ij}}$ computation of $\delta_1$ and $\delta_3$ take the form of (64). Similarly, by (59) and (61), computation of $\delta_5$ takes the form of (64), as well. Moreover, computation of $\delta_2$ and $\delta_4$ 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 $O(n^3)$ 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 the problem (51)) to $\rho_n^2$ 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 $H_2$ 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
the $H_2$-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