Feedforward Estimators for the Distributed Average Tracking of Bandlimited Signals in Discrete Time with Switching Graph Topology

Bryan Van Scoy, Randy A. Freeman, and Kevin M. Lynch

***Abstract*— We consider the distributed average tracking problem where a group of agents estimates the global average of bandlimited signals using only local communication. An estimator is designed to solve this problem with minimal error. Previous discrete-time designs are limited to tracking signals which either are constant, are slowly varying, have a known model (or frequency), or consist of a single unknown frequency which can be estimated. In contrast, we propose a feedforward design which is capable of tracking the average of arbitrary bandlimited signals. The communication graph is assumed to be connected and symmetric with non-zero weighted Laplacian eigenvalues in a known interval, although simulations show that the performance degrades gracefully as these assumptions are violated. Our design also provides the estimate of the average without delay and is robust to changes in graph topology.**

I. INTRODUCTION

The distributed average tracking problem consists of a group of agents where each agent uses local communication with its network neighbors along with a local estimator to estimate the average input of all the agents. The input signals are assumed to be time-varying with a known band limit. Many applications in the decentralized control of multi-agent systems require estimators which solve the distributed average tracking problem and are robust to changes in the communication network [1], [2], [3], [4], [5].

Robustness to changes in network topology is an important property of a distributed estimator, as network changes can occur due to

- mobile agents with range-limited communication,
- dropped packets,
- agent failure, and
- addition of agents to the network.

Distributed average tracking is sometimes referred to as the average consensus problem, which has been studied extensively [6], [7], [8], [9]. Most average consensus estimators, however, are designed for inputs which are either constant or slowly varying. We use the term distributed average tracking to refer to the average consensus problem when the input is time-varying.

The simplest average consensus estimator is given by [10]

$$x_{k+1} = (I - k_p L)x_k \tag{1}$$

This work is supported in part by the Office of Naval Research. The authors are with the Department of Electrical Engineering and Computer Science (Van Scoy and *Freeman) and the Department of Mechanical Engineering (*Lynch), Northwestern University, Evanston, IL 60208, USA. *Also member of Northwestern Institute on Complex Systems (NICO). Emails: bryanvanscoy2012@u.northwestern.edu, freeman@eecs.northwestern.edu, kmlynch@northwestern.edu.

where L is the graph Laplacian, $u = x_0$ is the vector of inputs, and $y_k = x_k$ is the output. In the limit as $k \to \infty$, y_k converges to the mean of u. This estimator does not solve the distributed average tracking problem, however, because the input enters the system only as the initial condition. The estimator calculates the average of the inputs, but is incapable of *tracking* the average of a time-varying input.

As a first step to tracking time-varying inputs, the average consensus estimator in (1) was modified so that the input enters as the input of a dynamical system, not as the initial state. Examples are the P and PI estimators [11]. Since the input is no longer the initial state, estimators are referred to as *robust to initial conditions* if the steady-state value is independent of the initial state. In this case, the estimator can track slowly-varying inputs and can recover from changes in graph topology after a transient without any reinitialization. These estimators are restricted to slowly-varying inputs, however, since they are designed to track constant inputs with zero steady-state error.

The estimators in [12] and [13] solve the discrete-time dynamic average tracking problem, but the estimators are not robust to the initial conditions. The systems must be initialized correctly when agents enter or leave the network which is not always possible in realistic scenarios.

To accommodate arbitrarily fast time-varying inputs with a known model (or frequency), the internal model estimator was introduced [9]. The internal model of the input is placed in the feedback loop causing the estimator to have zero steady-state error. Although this estimator can track arbitrarily fast time-varying signals, the model of the input must be known a priori to design the estimator, and no guarantees of steady-state error are given if the input does not identically match the model. In the case when the signal frequency is unknown, it was shown that the frequency can be estimated and the estimate used in the internal model estimator [14]. This method can achieve zero steady-state error, but the signal must be composed of a single frequency.

The internal model estimator can track the global average after changes in the graph, but only after a transient. The size of the transient can be reduced by applying a lowpass post-filter [15], although it is unclear how the filter should be designed to minimize the transient.

All previously mentioned estimators for distributed average tracking use feedback designs. In this paper, we design a discrete-time feedforward estimator to solve the distributed average tracking problem. The input signals are assumed to be arbitrary bandlimited signals with a known cutoff frequency, which can be achieved using lowpass input filters.

The output of each agent tracks the global average of all input signals with bounded error and no delay, i.e., the estimate of the global average is known on each agent at every iteration and it approximates the average of the inputs at that time instant. The communication graph is assumed to be connected and undirected at each iteration with known bounds on the non-zero eigenvalues of the Laplacian matrix. The graph is allowed to change at each iteration without affecting the performance of the estimator, so long as the graph satisfies the assumptions at each iteration. It is also shown through simulations that the performance degrades gracefully as the assumptions on the graph are violated.

II. DISTRIBUTED AVERAGE TRACKING

Consider a group of N agents where each agent has a local scalar input signal. The input at time k on agent i is denoted u_k^i. Each agent runs an estimator using its own local input along with information from its neighbors to produce a local scalar output signal y_k^i. The distributed average tracking problem is for the output of each agent to track the global average of the inputs. We define the error on agent i at time k to be the absolute difference between the output of the agent and the global average at the same time instant,

$$e_k^i = \left| y_k^i - \frac{1}{N}\sum_{i=1}^{N} u_k^i \right|.$$

The communication topology is modeled as a weighted undirected graph G. Define the adjacency matrix of G to be $A = [a_{ij}] \in \mathbb{R}^{N\times N}$ where $a_{ij} = a_{ji} > 0$ if agents i and j can communicate and zero otherwise (with $a_{ii} = 0$). The neighbors of agent i, denoted $\mathcal{N}_i$, is the set of agents with which agent i can communicate. The number of agents in $\mathcal{N}_i$ is the degree of agent i, denoted $\deg(i)$. Define the $N \times 1$ vectors 1_N and 0_N of all ones and zeros, respectively, and I as the $N \times N$ identity matrix. Then the Laplacian matrix is $L = \text{diag}(A1_N) - A$ which is symmetric positive semidefinite and satisfies $L1_N = 0_N$.

Let λ_{min} and λ_{max} with $0 < \lambda_{\text{min}} \le \lambda_{\text{max}}$ denote lower and upper bounds on the non-zero eigenvalues of L so that the eigenvalues of L are in $\{0\} \cup [\lambda_{\text{min}}, \lambda_{\text{max}}]$. The graph is connected if and only if the zero eigenvalue has multiplicity one. We can decompose the Laplacian as $L = V\Lambda V^T$ where $V = [v_1, \ldots, v_N]$, $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N)$, and $VV^T = V^TV = I$. For $i = 1, \ldots, N$, v_i is an eigenvector of L with eigenvalue λ_i.

The weights a_{ij} can be chosen to optimize the convergence rate when the graph topology is known [16]. When the graph is unknown, however, it is often useful to choose a weighting scheme which bounds the eigenvalues of the Laplacian. For example, the decentralized weighting scheme $a_{ij} = 1/[\deg(i) + \deg(j)]$ restricts the eigenvalues of L to the interval $[0, 1]$ which allows us to use $\lambda_{\text{max}} = 1$ when the graph is unknown [17].

Dynamic estimators are often referred to as robust if they are robust to the initial conditions. In this case, changes in the graph topology can cause a transient in the estimate of the average, but the global average is recovered once the transient decays. If the graph topology changes quickly enough, the estimator remains in the transient state and cannot track the global average. We now define a stronger robustness property. Estimators with this property can track the average even when the graph topology changes quickly.

Definition 1 (Robust to changes in graph topology): Let $\mathcal{L}$ be a set of graph Laplacians. Consider implementing an estimator in the following two scenarios:

1) The graph is fixed with Laplacian $L \in \mathcal{L}$.
2) The graph is allowed to change at each iteration with the graph Laplacian at time k given by L_k where $L_k \in \mathcal{L}$ for $k = 1, 2, \ldots$.

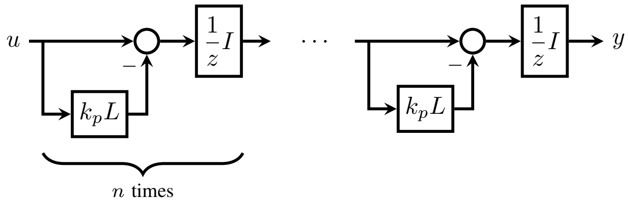
An estimator is said to be *robust to changes in graph topology* when the steady-state error using the worst-case sequence of graphs in (2) is equal to the steady-state error using the worst-case constant graph in (1).

III. FEEDFORWARD ESTIMATOR DESIGN

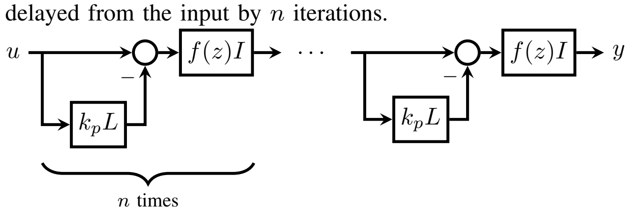
Instead of using feedback as in equation (1), we propose the use of feedforward estimators to solve the distributed average tracking problem. As a first approach, consider the estimator shown in Figure 1a which simply applies the consensus matrix $I - k_pL$ to the input n times. Each iteration requires time since multiplication by L requires communication among neighbors, so there is a delay of $1/z$ between each iteration. This design works well for *any* inputs (not just bandlimited inputs) and is robust to changes in graph topology. The drawback, however, is that the output is delayed by n steps from the input, so achieving a better estimate requires more delay.

To fix the delay, we could replace $1/z$ with a filter $f(z)$ as shown in Figure 1b. The frequency response of the filter $f(z)$ should be designed to approximate unity for inputs with frequencies in $[-\theta_c, \theta_c]$ where θ_c is the cutoff frequency of the input signals. Then the transfer function approximates $(I - k_pL)^n$ in $[-\theta_c, \theta_c]$. We could choose $f(z) = 1$, but this results in an n-hop estimator. In this case, each iteration would require n rounds of communication to be done sequentially since the result of each round is needed for the next. This is due to Laplacian blocks being directly connected in the block diagram without any delay between them. To prevent this, we require $f(z)$ to be strictly proper so that there is no direct feedthrough between Laplacian blocks. The estimator can then be implemented in one-hop meaning that each agent can broadcast all of its information in a single packet at each iteration. This fixes the delay problem, but the estimator is not robust to changes in graph topology.

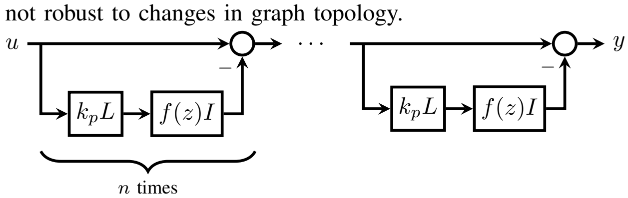
The design in Figure 1b filters both the consensus and disagreement directions. Since we only require a strictly proper filter between consecutive Laplacian blocks, we could also place the filter directly after each Laplacian block as in Figure 1c. A benefit of this design is that it can handle any signals (not only bandlimited signals) which are common to all agents since the consensus direction is a straight wire from the input to the output. This estimator also has no delay at the output, but is still not robust to changes in graph



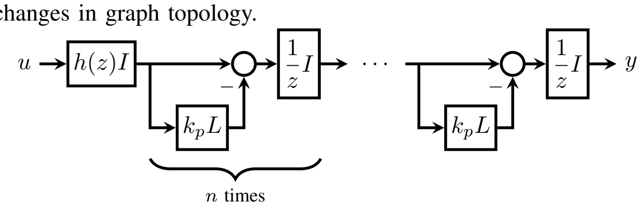
(a) Estimator consisting of n steps of standard average consensus. The estimator is robust to changes in graph topology, but the output is delayed from the input by n iterations.



(b) Estimator with the filter $f(z)$ in both the consensus and disagreement directions. The output is not delayed, but the estimator is not robust to changes in graph topology.



(c) Estimator with the filter $f(z)$ only in the disagreement directions. The output is not delayed, but the estimator is not robust to changes in graph topology.



(d) Estimator where the filter $h(z) = [zf(z)]^n$ is implemented *before* passing through the graph Laplacian. The output is not delayed and the estimator is robust to changes in graph topology.

Fig. 1: Block diagrams of the proposed feedforward estimators. Each marked section is repeated in series n times.

topology. When the graph is fixed, however, simulations in Section IV indicate that this design can achieve smaller steady-state error compared to the other designs.

To see why the estimators in Figures 1b and 1c are not robust to changes in graph topology, consider implementing the estimators on a switching graph. Changes in the graph cause the eigenvalues to change, and random changes in the eigenvalues create high-frequency components in the signal after passing through the Laplacian. These high-frequency components are not in the passband of $f(z)$ and are therefore amplified when filtered through $f(z)$ (see Section III-A) causing the error to be large.

We can make an estimator with the same transfer function as that in Figure 1b which *is* robust to changes in graph topology by applying the filter $f(z)$ before the signal passes through the Laplacian as shown in Figure 1d. After the pre-filter $[zf(z)]^n$, the rest of the estimator is identical to that in Figure 1a which works for any inputs (not just bandlimited inputs) and is robust to changes in graph topology. The problem with the estimator in Figure 1a was that the output was delayed, but this is offset in the estimator in Figure 1d by the pre-filter. Therefore, we propose the estimator in Figure 1d to solve the distributed average tracking problem without delay and over graphs with changing topology.

The filter $f(z)$ is not a standard lowpass filter, although we give a design procedure in Section III-A. For now, we assume that we know the maximum deviation of $f(e^{j\theta})$ from unity for $\theta \in [-\theta_c, \theta_c]$, given by

$$\delta := \max_{\theta \in [-\theta_c, \theta_c]} \left|1 - f(e^{j\theta})\right|, \tag{2}$$

and the infinity norm of $f(z)$,

$$||f||_\infty = \max_{\theta \in [-\pi, \pi]} \left|f(e^{j\theta})\right|. \tag{3}$$

The transfer function from the input to the output of the estimator in Figure 1d is

$$H(z, L) = [(I - k_pL)f(z)]^n = V\left[(I - k_p\Lambda)f(z)\right]^n V^T,$$

and the transfer function from the input to the error is

$$H_{\text{err}}(z, L) = v_1 v_1^T - V\left[(1 - k_p\Lambda)f(z)\right]^n V^T \tag{4}$$

where $v_1 = 1_N/\sqrt{N}$. The singular values of $H_{\text{err}}(z, L)$ are

$$\sigma = \begin{cases} 1 - f^n(z), & \lambda = 0 \\ -\left[(1 - k_p\lambda)f(z)\right]^n, & \lambda \in [\lambda_{\text{min}}, \lambda_{\text{max}}]. \end{cases} \tag{5}$$

The maximum singular value of the error transfer function is defined as

$$\sigma_{\text{max}} := \max_{\substack{\lambda \in \{0\} \cup [\lambda_{\text{min}}, \lambda_{\text{max}}] \\ \theta \in [-\theta_c, \theta_c]}} \sigma \tag{6}$$

where the non-zero graph eigenvalues are viewed as an uncertain parameter λ and the frequencies of the input signal are viewed as an uncertain parameter θ. This is the maximum singular value over both the consensus direction and disagreement directions. The error of the estimator can be bounded using σ_{max} and the size of the input,

$$||e||_\infty \leq ||e||_2 \leq \sigma_{\text{max}}||u||_2 \leq \sqrt{N}\sigma_{\text{max}}||u||_\infty. \tag{7}$$

To minimize the error, we want to minimize σ_{max}. The following lemma gives an upper-bound on σ_{max}. The proof is omitted for brevity.

Lemma 1: Consider the estimator in Figure 1d with $k_p = 2/(\lambda_{\text{min}} + \lambda_{\text{max}})$. For all bandlimited input signals with cutoff frequency θ_c and all connected undirected graphs with non-zero Laplacian eigenvalues in $[\lambda_{\text{min}}, \lambda_{\text{max}}]$, the maximum singular value of the error transfer function $H_{\text{err}}(z, L)$ in (4) is bounded by

$$\sigma_{\text{max}} \leq \max\left\{(1+\delta)^n - 1, \left[(1+\delta)\frac{1-\lambda_r}{1+\lambda_r}\right]^n\right\} \tag{8}$$

where δ is given by (2) and $\lambda_r = \lambda_{\text{min}}/\lambda_{\text{max}}$.

Note that σ_{max} is a function of δ (how close $f(z)$ approximates unity in $[-\theta_c, \theta_c]$), λ_r (the ratio of Laplacian eigenvalues), and n (the size of the estimator). To minimize the error, we want to design $f(z)$ to minimize $\sigma_{\max}$. However, this optimization causes a large transient due to the initial conditions (see the simulations in Section IV). Implementing the estimator on a computer with finite precision introduces quantization noise. If the transient becomes too large, the input cannot be distinguished from the quantization noise so the estimator cannot track the signal. To fix this issue, we upper-bound $||H||_\infty$ by a constant $H_\infty^{\max}$ depending on the precision of the arithmetic used to implement the estimator which ensures that the transient due to the initial conditions is not too large. A bound on the infinity norm of the estimator $H(z, L)$ is

$$||H||_\infty = \max_{\theta\in[-\pi,\pi]} |f^n(e^{j\theta})| \leq ||f||_\infty^n. \tag{9}$$

Problem 1: Given bounds on the non-zero eigenvalues of the Laplacian, the cutoff frequency of the inputs, and the maximum allowable gain of the estimator, choose the size of the estimator and the pre-filter to minimize the worst-case gain from the input to the error subject to the gain from the input to the output not exceeding the maximum allowable gain. That is, solve

$$\min_{n,f(z)} \sigma_{\max} \quad \text{s.t. } ||H||_\infty \leq H_\infty^{\max}. \tag{10}$$

A. Design of $f(z)$

The design of the feedforward estimator requires a filter $f(z)$ with the following properties:

- $f(z)$ is strictly proper
- $f(e^{j\theta}) \approx 1$ for $\theta \in [-\theta_c, \theta_c]$.

We need the filter to approximate unity in both magnitude *and* phase in the passband, so a standard lowpass filter cannot be used. Instead, we set

$$f(z) = 1 - \frac{g(z)}{\lim\limits_{z\to\infty} g(z)} \tag{11}$$

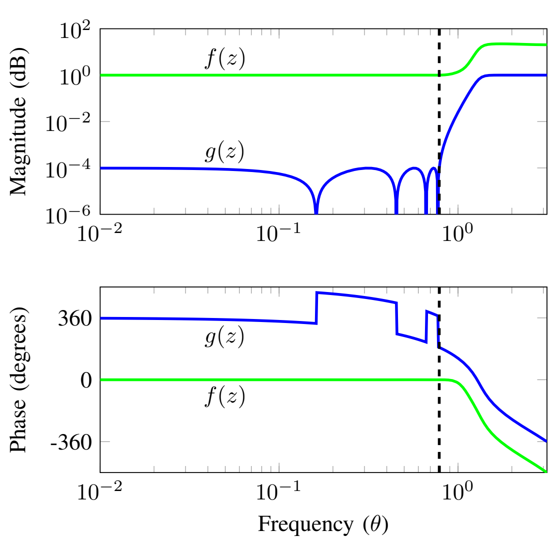


Fig. 2: Bode plot of $g(z)$ (blue) and $f(z)$ (green) with $m = 8$, $\epsilon = 10^{-4}$, and $\theta_c = \pi/4$. The vertical line indicates θ_c.

where $g(z)$ is a filter to be designed which approximates zero in the passband. Then $\lim_{z\to\infty} f(z) = 0$ so $f(z)$ is strictly proper. To minimize the distance from unity, $g(z)$ is chosen to be a Type II Chebyshev highpass filter which equioscillates in the stopband (which is $[-\theta_c, \theta_c]$). The transfer function of the Type II Chebyshev highpass filter is obtained as follows. First, the corresponding continuous filter $G(s)$ is obtained. The continuous filter is the unique stable filter with frequency response

$$|G(j\omega)| = \frac{\epsilon T_m(\omega/\omega_0)}{\sqrt{1+\epsilon^2 T_m^2(\omega/\omega_0)}} \tag{12}$$

where T_m is the degree m Chebyshev polynomial of the first-kind, ω_0 is the cutoff frequency, and ϵ is a design parameter. The s-plane poles are the left-half plane roots of the denominator of (12), and the zeros are the roots of the numerator with multiplicity one. These are given by $P_i = j\omega_0 \cos(\theta_i - j\gamma)$ and $Z_i = j\omega_0 \cos(\theta_i)$ for $i = 1, \ldots, m$ where $\theta_i = \pi(2i-1)/(2m)$, $\gamma = \text{asinh}(1/\epsilon)/m$, and $j = \sqrt{-1}$ is the imaginary unit. The z-plane poles and zeros are then obtained using the bilinear transform, $p_i = (2+P_i)/(2-P_i)$ and $z_i = (2+Z_i)/(2-Z_i)$. The gain is chosen such that the transfer function is unity at $z = -1$, so $K = \prod_{i=1}^m (1+z_i)/(1+p_i)$. Since the bilinear transform warps the frequencies, we take $\omega_0 = 2\tan(\theta_c/2)$ so that the cutoff frequency of the discrete filter is θ_c. The transfer function of the discrete Type II Chebyshev highpass filter is then

$$g(z) = K \prod_{i=1}^m \frac{z - z_i}{z - p_i}. \tag{13}$$

The magnitude of the filter response oscillates between 0 and $\epsilon/\sqrt{1+\epsilon^2}$ in the stopband, so

$$\max_{\theta\in[-\theta_c,\theta_c]} \left|g(e^{j\theta})\right| = \frac{\epsilon}{\sqrt{1+\epsilon^2}}. \tag{14}$$

Then $f(z)$ is given by

$$f(z) = 1 - \frac{g(z)}{K} = 1 - \prod_{i=1}^m \frac{z - z_i}{z - p_i}. \tag{15}$$

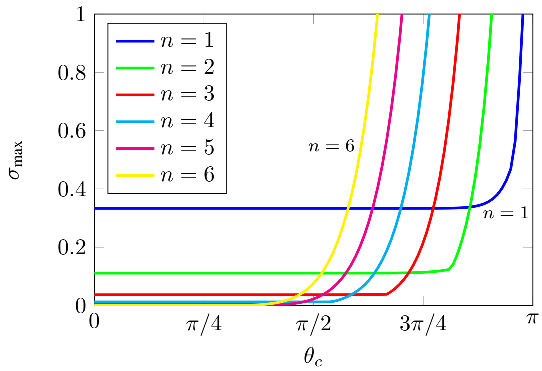
Example Bode plots of $g(z)$ and $f(z)$ are given Figure 2. Note that $f(z)$ is strictly proper and approximates unity in the passband with maximum error

$$\delta := \max_{\theta\in[-\theta_c,\theta_c]} \left|1 - f(e^{j\theta})\right| = \frac{1}{K} \frac{\epsilon}{\sqrt{1+\epsilon^2}}. \tag{16}$$

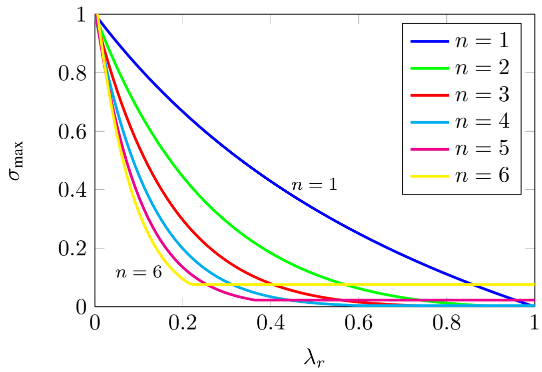
Since $||g||_\infty = 1$, a bound on the infinity norm of $f(z)$ is $||f||_\infty \leq 1 + 1/K$. The optimization problem in equation (10) then becomes

$$\begin{aligned} &\min_{n,m,\epsilon} \max \left\{ (1+\delta)^n - 1, \left[(1+\delta)\frac{1-\lambda_r}{1+\lambda_r} \right]^n \right\} \\ &\text{s.t. } \left(1 + \frac{1}{K}\right)^n \leq H_\infty^{\max}. \end{aligned} \tag{17}$$

We now show how to approximate the solution to this problem. Consider fixing n. Then the objective function is minimized by making δ small, which is equivalent to making



(a) $\lambda_r = 0.5$



(b) $\theta_c = \pi/2$

Fig. 3: Plot of $\sigma_{\max}$ as a function of θ_c and λ_r for $n = 1, 2, \ldots, 6$.

ϵ small and K large, while the constraint only requires K to be large enough. Therefore, we approximate the solution to (17) for fixed n by solving

$$\min_{m,\epsilon} \quad \epsilon \quad \text{s.t.} \quad K \geq \frac{1}{\sqrt[n]{H_\infty^{\max}} - 1}. \tag{18}$$

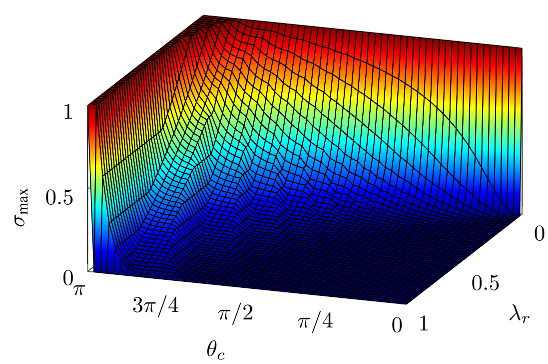
Note that K is an increasing function of ϵ and m. The optimal solution to (18) is obtained as $m \to \infty$. This is not practical, however, since m is the order of the filter $f(z)$. Therefore, we consider some maximum allowable filter order $m_{\max}$ and set $m = m_{\max}$. Bisection is then performed on ϵ to find the ϵ for which the constraint is satisfied with equality. This approximately solves the optimization problem (18) for fixed n, and the resulting values of $\sigma_{\max}$ are shown in Figure 3. We solve this problem for $n = 1, 2, \ldots$ and take the smallest value to be the optimum. The results are shown in Figure 4.

IV. SIMULATIONS

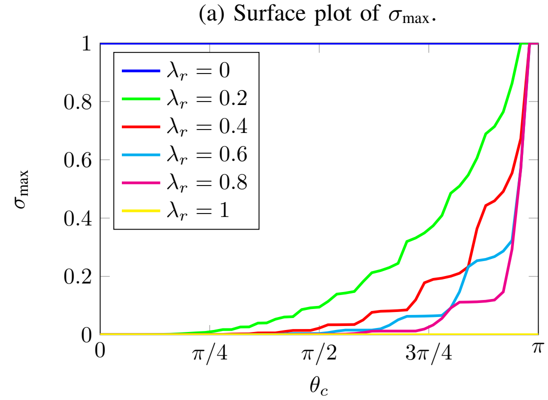
The proposed estimator design is simulated using a 500-node undirected geometric random graph. The graph Laplacian is constructed using inverse degree weighting and has non-zero eigenvalues in $[0.1240, 0.5989]$ so that $\lambda_r = 0.2071$. Each input signal is bandlimited with cutoff frequency $\theta_c = \pi/4$. The Fourier transform of each input signal is constructed as follows: random complex values are assigned to the part of the frequency spectrum corresponding to $\theta \in [0, \theta_c]$, the complex conjugate values are assigned to $\theta \in [-\theta_c, 0]$ (so that the input signal is real), and the spectrum in $\theta \in [-\pi, -\theta_c] \cup [\theta_c, \pi]$ is set to zero. The input signal is then obtained using the inverse Fourier transform. The degree of $f(z)$ is chosen to be $m = 8$. Double precision arithmetic is used, so we take $H_\infty^{\max} = 2 \times 10^{13}$ which ensures that the transient does not become too large.

The error of the estimator is calculated using the maximum absolute error over all agents normalized by the maximum size of the global average,

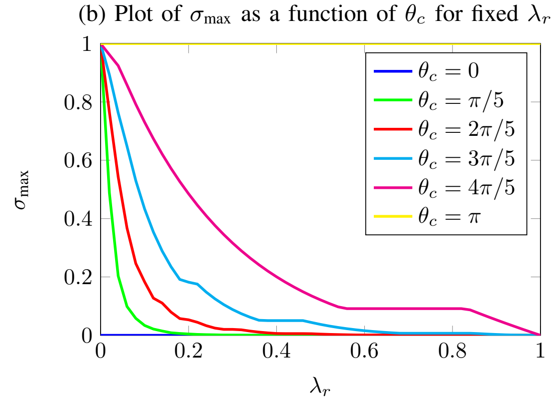
$$e_k = \frac{\max_{i=1,\ldots,N} \left| y_k^i - \bar{u}_k \right|}{\max_j |\bar{u}_j|} \qquad \text{where} \qquad \bar{u}_k = \frac{1}{N} \sum_{i=1}^{N} u_k^i.$$



(a) Surface plot of $\sigma_{\max}$.



(b) Plot of $\sigma_{\max}$ as a function of θ_c for fixed λ_r.



(c) Plot of $\sigma_{\max}$ as a function of λ_r for fixed θ_c.

Fig. 4: Upper-bound on $\sigma_{\max}$ as a function of θ_c and λ_r using $H_\infty^{\max} = 2 \times 10^{13}$ and $m_{\max} = 8$.

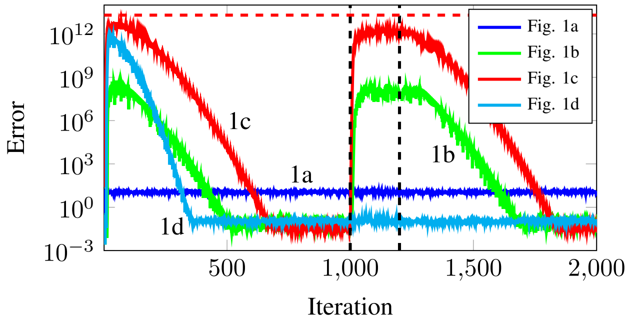


Fig. 5: Error for each estimator in Figure 1. The horizontal red dashed line is $H_\infty^{\max}$. Between the two vertical black dashed lines, packets are dropped at each iteration with probability 0.1.

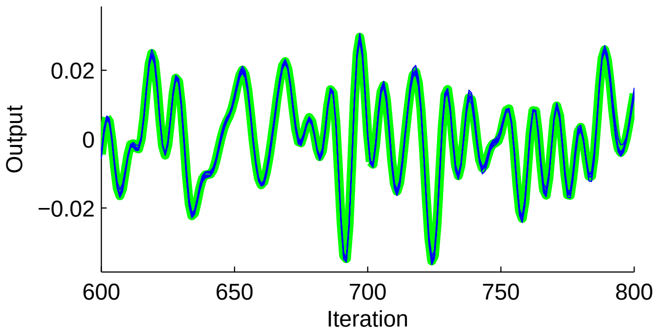


Fig. 6: Output of each agent for the estimator in Fig. 1d (blue), and the global average of all inputs (green). The time scale is chosen such that the estimator is in steady-state. The graph is constant throughout the simulation.

Figure 5 shows the error for each estimator in Figure 1. Packets are dropped independently at each iteration with probability 0.1 between iterations $k = 1000$ and $k = 1200$. The output of the estimator in Figure 1a is delayed, so it cannot track the average of the time-varying input signals. The estimators in Figures 1b and 1c achieve small steady-state error when the graph is constant, but are not robust to changes in graph topology and have large error whenever the graph changes. They return to tracking the global average after the graph returns to being constant, but only after a long transient. Finally, the estimator in Figure 1d tracks the global average and is robust to changes in graph topology. The error is slightly higher when packets are dropped, but this is due to the graph not satisfying the assumptions at each iteration. The output of the agents for this estimator compared to the global average is shown in Figure 6.

One benefit of the estimator in Figure 1d is that the performance degrades gracefully when the assumptions on the graph are violated as illustrated in Figure 7. The estimator is designed using a larger eigenvalue range than that of the graph to increase the performance when the graph topology changes. Packets are dropped randomly at each iteration with probabilities $p = 0$, 0.25, and 0.5. Due to the dropped packets, the graph at each iteration can be directed, disconnected, and/or have eigenvalues outside of the designed range which

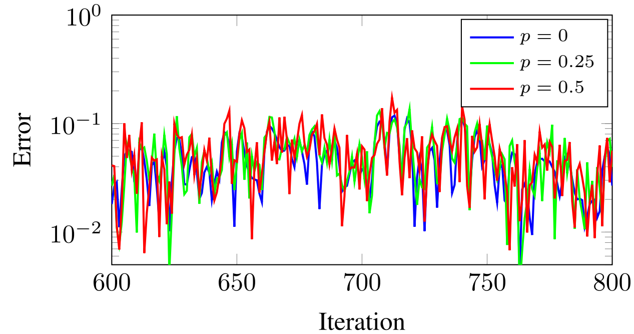


Fig. 7: Steady-state error using the estimator in Figure 1d. At each iteration, packets are dropped with probability p. The estimator is designed using $\lambda_{\text{min}} = 0.0248$ and $\lambda_{\text{max}} = 1$.

violates the assumptions. Even with the assumptions of the graph being violated at some iterations, the estimator still tracks the global average with no more than about 10% error.

References

[1] P. Yang, R. Freeman, and K. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[2] J. Cortes, "Distributed kriged kalman filter for spatial estimation," *IEEE Trans. Autom. Control*, vol. 54, no. 12, pp. 2816–2827, 2009.

[3] R. Aragues, J. Cortes, and C. Sagues, "Distributed consensus on robot networks for dynamically merging feature-based maps," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 840–854, 2012.

[4] C. K. Peterson and D. A. Paley, "Distributed estimation for motion coordination in an unknown spatially varying flowfield," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 894–898, 2013.

[5] K. Lynch, I. Schwartz, P. Yang, and R. Freeman, "Decentralized environmental modeling by mobile sensor networks." in *IEEE Trans. on Robotics*, 2008.

[6] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," *IFAC World Congress*, pp. 1–6, 2005.

[7] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3169–3174, Dec. 2012.

[8] S. Kia, J. Cortes, and S. Martinez, "Singularly perturbed algorithms for dynamic average consensus," in *Proc. of the 2013 European Control Conf.*, July 2013, pp. 1758–1763.

[9] H. Bai, R. Freeman, and K. Lynch, "Robust dynamic average consensus of time-varying inputs," in *Proc. of the 49th IEEE Conf. on Decision and Control*, 2010, pp. 3104–3109.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. of the 42nd IEEE Conf. on Decision and Control*, vol. 5, Dec 2003, pp. 4997–5002.

[11] R. Freeman, P. Yang, and K. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. of the 45th IEEE Conf. on Decision and Control*, 2006, pp. 338–343.

[12] M. Zhu and S. Martnez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322 – 329, 2010.

[13] S. S. Kia, J. Cortes, and S. Martinez, "Dynamic Average Consensus under Limited Control Authority and Privacy Requirements," *ArXiv e-prints*, Jan. 2014.

[14] H. Bai, "Adaptive motion coordination with an unknown reference velocity," in *Proc. of the 2015 Amer. Control Conf.*, July 2015, pp. 5581–5586.

[15] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "Asymptotic mean ergodicity of average consensus estimators," in *Proc. of the 2014 Amer. Control Conf.*, June 2014, pp. 4696–4701.

[16] P. Yang, R. Freeman, and K. Lynch, "Optimal information propagation in sensor networks," in *Proc. of the 2006 IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 3122–3127.

[17] R. Freeman, T. Nelson, and K. Lynch, "A complete characterization of a class of robust linear average consensus protocols," in *Proc. of the 2010 Amer. Control Conf.*, 2010, pp. 3198–3203.