A Fast Robust Nonlinear Dynamic Average Consensus Estimator in Discrete Time ⋆

Bryan Van Scoy * **Randy A. Freeman** * **Kevin M. Lynch** **

* *Elec. Eng. & Comp. Sci. Dept., Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208 USA (e-mail: bryanvanscoy2012@u.northwestern.edu, freeman@eecs.northwestern.edu).*
** *Mech. Eng. Dept., Northwestern University, 2145 Sheridan Rd., Evanston, IL 60208 USA (e-mail: kmlynch@northwestern.edu).*

Abstract: We present a new discrete-time dynamic average consensus estimator which has significant performance advantages over existing designs. It uses nonlinear oscillators to achieve both robustness and internal boundedness. Its primary disadvantage is that it cannot generally track averages of unbounded inputs with bounded error.

Keywords: Average consensus, synchronization of oscillators, optimal robust estimator design.

1. INTRODUCTION

We consider the average consensus problem in which each node in a graph computes the global average of node inputs using data collected from its neighbors. This problem appears in a number of applications involving sensor networks, including environmental monitoring (Olfati-Saber, 2005, 2007; Cortés, 2009; Lynch et al., 2008; Bai et al., 2011; Peterson and Paley, 2013), formation control (Yang et al., 2008), sensor fusion (Olfati-Saber and Shamma, 2005), and mapping (Aragüés et al., 2012). In this paper we consider diffusive approaches to this problem which are inherently scalable, distributed, indifferent to network structure, and frugal in their use of memory, computation, and communication resources.

If the node inputs are constants, then their average can be computed using a variety of diffusive methods in which node outputs converge to the average of the initial node states. Such methods are known as static average consensus and have been studied extensively since Tsitsiklis (1984). Spanos et al. (2005) introduced dynamic average consensus in which the node inputs appear explicitly in the update equations themselves, not just as initial states. In the dynamic case, node inputs can change with time without the need for any reinitialization, and each node maintains a running local estimate of the current global average input. Hence we say that the dynamic average consensus algorithm running on each node is an *estimator*.

In this paper we focus our attention on dynamic average consensus in discrete time. We present a new estimator based partially on the nonlinear oscillator static consensus method in (Khan, 2009), but in dynamic form and with additional filters for improved convergence rates. Our estimator has a number potential advantages over existing dynamic average consensus estimators in discrete time, and in particular is the only diffusive estimator we know of which has all of the following properties:

Exactness: Under constant inputs, all node outputs converge to the exact global average in forward time.
Initialization robustness: The initial values of the internal node states have no effect on the steady-state estimates (provided they lie in some open region of attraction).
Time invariance: The estimator dynamics are time invariant (in particular, transient characteristics like settling time do not change as the estimator runs).
Internal stability: The estimator dynamics are bounded-input, bounded-state stable.
Fast convergence: The estimates converge to steady state at rates comparable to the fastest known static average consensus methods.
Single-word local broadcast: At each discrete-time update, each node broadcasts a single real value to its immediate neighbors.

Exactness is desirable not only because it guarantees zero steady-state errors for constant inputs, but also because it leads to small steady-state errors for slowly-varying inputs. The estimator in (Moallemi and Van Roy, 2006) is not exact and requires a design trade-off between the size of steady-state errors under constant inputs and the convergence rate. Initialization robustness makes it easy for the network to recover from changes in the network topology, temporary asymmetries in communication, or the addition or deletion of nodes; indeed, no special action is required. The estimators in (Zhu and Martínez, 2010; Yuan et al., 2012; Montijano et al., 2014b) lack initialization robustness and consequently must take care to correctly handle these situations without introducing additional errors in steady state. For example, Zhu and Martínez (2010) suggest that a node send a particular message to its neighbors upon leaving the network, but an abruptly failing node is unlikely to send such a message. Estimators having initialization robustness can also have a

⋆ This work was supported in part by a grant from the Office of Naval Research.

desirable ergodicity property which allows nodes to recover the value of the global average under switching network topologies (Van Scoy et al., 2014). The estimator in (Montijano et al., 2014a) is time varying, and in particular its settling times due to state perturbations grow without bound in forward time (so that the longer it runs, the slower it recovers from such perturbations). Finally, our previous designs in (Freeman et al., 2010; Bai et al., 2010; Elwin et al., 2013, 2014; Van Scoy et al., 2014) require multi-word local broadcast and generally do not converge as quickly as the fast static average consensus methods in (Ghosh et al., 1998; Kokiopoulou and Frossard, 2009; Oreshkin et al., 2010; Liu and Morse, 2011; Erseghe et al., 2011).

2. DYNAMIC AVERAGE CONSENSUS

We consider a simple undirected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ having a nonempty set $\mathcal{V}$ of nodes and a set $\mathcal{E}$ of edges, where each edge is an unordered pair of distinct nodes. We label and number the nodes and edges, writing $\mathcal{V} = \{1, \dots, n\}$ and $\mathcal{E} = \{\mathfrak{e}_1, \dots, \mathfrak{e}_m\}$, where $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$. Let $B \in \{-1, 0, 1\}^{n \times m}$ be the oriented incidence matrix for $\mathbb{G}$ defined as

$$B_{i\ell} = \begin{cases} 1 & \text{if } i \in \mathfrak{e}_\ell \text{ and } i = \min\{j \,:\, j \in \mathfrak{e}_\ell\} \\ -1 & \text{if } i \in \mathfrak{e}_\ell \text{ and } i = \max\{j \,:\, j \in \mathfrak{e}_\ell\} \\ 0 & \text{if } i \notin \mathfrak{e}_\ell \,. \end{cases} \tag{1}$$

If we assign a positive weight $w_\ell > 0$ to each edge $\mathfrak{e}_\ell \in \mathcal{E}$ and form the $m \times m$ weight matrix $W = \operatorname{diag}\{w_1, \dots, w_m\}$, then $L = BWB^T$ is the $n \times n$ symmetric weighted Laplacian matrix for the graph $\mathbb{G}$. Because the column sums of B are zero, we have $L\mathbf{1} = 0$, where $\mathbf{1} \in \mathbb{R}^n$ denotes the column vector of all ones. Thus $\mathbf{1}$ is an eigenvector corresponding to a zero eigenvalue of L, and one can show that if $\mathbb{G}$ is connected then the remaining $n-1$ eigenvalues of L are all positive.

We assume knowledge of an upper bound $\lambda_{\max}$ on $\|L\|$. This knowledge will typically come from the scheme for choosing the edge weights. For example, if the weight w_ℓ for each edge $\mathfrak{e}_\ell = \{i, j\}$ is

$$w_\ell = \frac{1}{\deg(i) + \deg(j)} \,, \tag{2}$$

where $\deg(i)$ denotes the degree of node i, then $\|L\|$ will be no larger than $\lambda_{\max} = 1$ (Freeman et al., 2010). This upper bound is tight in the sense that $\|L\| = 1$ for some graphs using these weights (e.g., the graph having just two nodes connected by an edge). We also assume knowledge of a "soft" positive lower bound $\lambda_{\min}$ on the nonzero eigenvalues of L. This bound is soft in the sense that if it is violated, then our estimators will still converge but without any guarantees on their rates of convergence. We accept such a soft bound as it may difficult to obtain a non-conservative hard lower bound on the nonzero eigenvalues of L. As we will see below, the convergence rates of our estimators will depend on the ratio $\lambda_r = \lambda_{\min}/\lambda_{\max}$. If L is known and weights can be chosen in a centralized manner, then one can choose them to maximize the ratio λ_r. This is a convex optimization problem essentially equivalent to the one formulated by Xiao and Boyd (2004).

Suppose each node i maintains an internal scalar state $x_i(k)$ at each discrete time $k \in \mathbb{N}$, which it updates as follows:

$$x_i(k+1) = x_i(k) - h \sum_{\{i,j\} = \mathfrak{e}_\ell} w_\ell \big[x_i(k) - x_j(k)\big] \,, \tag{3}$$

where $h > 0$ is a constant step size and the sum is taken over all neighbors of i, that is, over all $j \in \mathcal{V}$ such that $\{i, j\} = \mathfrak{e}_\ell$ for some $\mathfrak{e}_\ell \in \mathcal{E}$. If we stack these states into a vector $x = [x_1 \ \dots \ x_n]^T$, then we can write the update more compactly as

$$x(k+1) = (I - hL)x(k) \,, \tag{4}$$

where I denotes the $n \times n$ identity matrix. It is well known that if $h\|L\| < 2$ and if $\mathbb{G}$ is connected, then as $k \to \infty$, all entries of the vector $x(k)$ converge to the average of the entries in the initial state vector $x(0)$. Variations on this standard iteration have been considered by many authors, at least as far back as Tsitsiklis (1984). An appropriate choice for the step size is

$$h = \frac{2}{\lambda_{\min} + \lambda_{\max}} \,. \tag{5}$$

Indeed, this choice maximizes the worst-case convergence rate in the sense that it minimizes the maximum modulus of the non-unity eigenvalues of $I - hL$ over all Laplacians L having nonzero eigenvalues in the interval $[\lambda_{\min}, \lambda_{\max}]$. The resulting maximum modulus is given by

$$\text{maximum modulus} = \frac{1 - \lambda_r}{1 + \lambda_r} \,, \tag{6}$$

where $\lambda_r = \lambda_{\min}/\lambda_{\max}$.

The system (4) is a static average consensus algorithm because its input is not a signal but a constant initial state. Indeed, Fig. 1 shows this system (4) as a block diagram in which the initial integrator state $x(0)$ appears explicitly as a constant input. In contrast, in dynamic average consensus, introduced by Spanos et al. (2005), each node has an input $u_i(k)$ which can change at each time step k, and the goal is for each node to maintain an estimate $y_i(k)$ of the global average

$$u_{\text{ave}}(k) = \frac{1}{n} \sum_{i=1}^{n} u_i(k) \,. \tag{7}$$

We can form a simple dynamic average consensus estimator by injecting the input into the static consensus block of Fig. 1, as shown in Fig. 2. Here $u = [u_1 \ \dots \ u_n]^T$ and $y = [y_1 \ \dots \ y_n]^T$ denote vectors of node inputs and outputs, respectively. In state-space form, the estimator in Fig. 2 is

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} I - hL & -hL \\ I & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \,. \tag{8}$$

Note that here we inject $u(k)$ at the output of the integrator block. In contrast, Zhu and Martínez (2010) inject the difference $u(k) - u(k-1)$ at the input to the integrator block, and in doing so they introduce an unnecessary delay together with a corresponding additional state per node. For this reason, we prefer the version (8) shown in Fig. 2.

One can show that if $h\|L\| < 2$, if $\mathbb{G}$ is connected, and if the inputs u_i are *constant*, then as $k \to \infty$, all entries of the output vector $y(k)$ in (8) converge to $x_{\text{ave}}(0) + u_{\text{ave}}$, where $x_{\text{ave}}(0)$ denotes the average of the entries in the initial state vector $x(0)$. In other words, the steady-state error vector under constant inputs is just $x_{\text{ave}}(0)\mathbf{1}$. Thus if $x_{\text{ave}}(0) = 0$, the steady-state error will be zero for constant inputs and small for slowly-varying inputs. Furthermore,

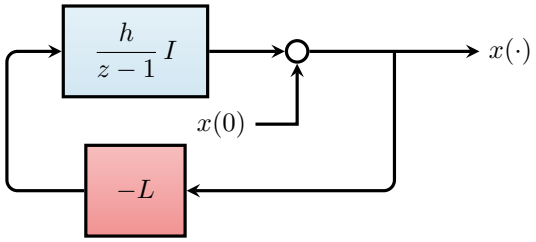

Fig. 1. The static average consensus estimator in (4).

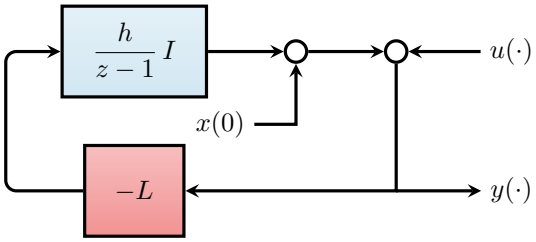

Fig. 2. A simple dynamic average consensus estimator.

this estimator can be cascaded to achieve zero steady-state error for inputs that are polynomial functions of the discrete time variable k: simply feed the output of the first stage into the input of the second, and so on as needed. As explained in (Zhu and Martínez, 2010), a cascade of s estimator stages will achieve zero steady-state error for polynomial inputs of degree up to $s-1$. However, this estimator in Fig. 2 lacks initialization robustness: its steady-state error depends on the value of $x_{\text{ave}}(0)$, which is not constant on any open set of initial states.

One way to achieve initialization robustness, developed by Montijano et al. (2014a), is to move the integrator pole in Fig. 2 initially away from $z = 1$ to a point strictly inside the unit circle in the complex plane, but then to let it slowly drift towards one as $k \to \infty$ using time-varying dynamics. However, this comes at a steep price: the settling times for the transients due to any new initialization errors increase without bound as $k \to \infty$. A different approach, developed by Bai et al. (2010), is to introduce a second Laplacian into the block diagram, but this also comes at a price: in some cases the resulting estimator converges considerably more slowly than the simple estimator in Fig. 2. In this paper we introduce a third method for achieving initialization robustness, one which keeps a single Laplacian in the loop but converges considerably more *quickly* than the simple estimator in Fig. 2. After we introduce our new estimator, we will discuss its advantages and disadvantages relative to the existing alternatives.

3. A ROBUST NONLINEAR ESTIMATOR

We must first understand why the simple estimator in Fig. 2 is not robust to initialization errors. In the consensus direction, the Laplacian block acts like an open circuit due to the fact that $L\mathbf{1} = 0$. When this block is open, the initial state $x(0)$ just adds to the output, thus producing steady-state errors which depend on $x(0)$. The problem does not appear in the disagreement directions (namely, directions orthogonal to $\mathbf{1}$) because the corresponding eigenvalues of $I - hL$ are strictly inside the unit circle, causing the contribution of any initial states to decay in

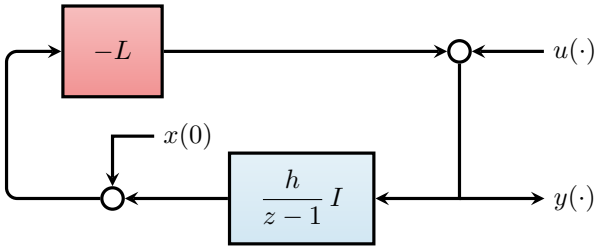

Fig. 3. A preliminary estimator obtained by swapping the order of the Laplacian and integrator blocks in Fig. 2.

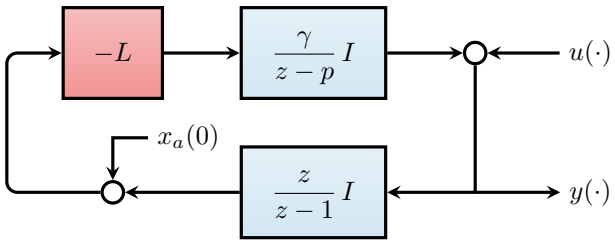

Fig. 4. Adding a filter to the estimator in Fig. 3 to allow time for communication.

these directions. One obvious way to fix this is to simply swap the order of the Laplacian and integrator blocks, leading to the estimator shown in Fig. 3. Now the initial integrator state $x(0)$ no longer appears at the output in the consensus direction because it passes through the Laplacian first; hence this (preliminary) new estimator is robust to initialization errors.

There are problems with the estimator in Fig. 3, however. The first problem is that now each node's output at time k depends on its neighbors' internal states at time k. This may not be practical, as it typically takes time for a node to collect information from its neighbors. Note that the original estimator in Fig. 2 avoids this problem because the output of its Laplacian block passes through the strictly proper integrator before reaching its output y. Thus to allow time for communication, we need a strictly proper block between the Laplacian and the output y, which we now insert as shown in Fig. 4. Here p is the pole of the added filter, and γ is a constant gain into which we have absorbed the step size h. The integrator block itself need not be strictly proper anymore (we only need one strictly proper block for a well-posed loop), so we add a zero in this block, putting it at $z = 0$ for reasons explained below. A state-space representation of the estimator in Fig. 4 is

$$\begin{bmatrix} x_a(k+1) \\ x_b(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} I & I & I \\ -\gamma L & pI - \gamma L & -\gamma L \\ 0 & I & I \end{bmatrix} \begin{bmatrix} x_a(k) \\ x_b(k) \\ u(k) \end{bmatrix}, \quad (9)$$

where now each node i has two internal states x_{ai} and x_{bi}, with $x_a = [x_{a1} \ \dots \ x_{an}]^T$ and $x_b = [x_{b1} \ \dots \ x_{bn}]^T$. Note that there is no need to account for the initial state vector $x_b(0)$ in the block diagram, as its contribution will decay in all directions (assuming a stable loop with $|p| < 1$).

To choose values for the parameters γ and p, we consider the root locus diagram for the loop, treating the Laplacian as a scalar parameter λ varying from $\lambda_{\min}$ to $\lambda_{\max}$. Because we added a zero at $z = 0$, the root locus for $0 < p < 1$ includes a circle of radius $\alpha = \sqrt{p}$, as shown in Fig. 5. To maximize the convergence rate, we would like to have

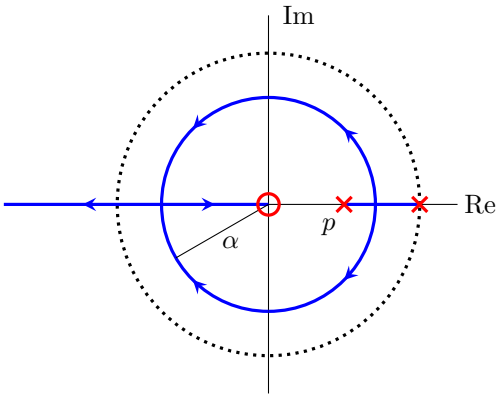

Fig. 5. Root locus of the loop in Fig. 4, which includes a circle of radius $\alpha = \sqrt{p}$.

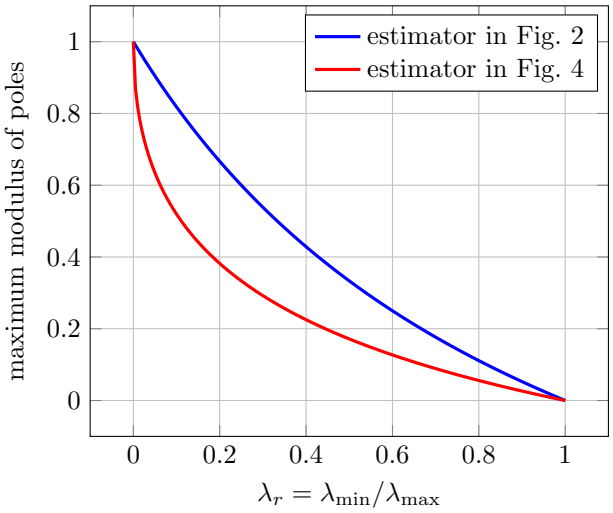

Fig. 6. A smaller maximum modulus of poles translates to a faster decay of the disagreement modes and thus faster estimator convergence.

all closed-loop poles within this circle while minimizing its radius α. We accomplish this by choosing α and γ so that there is a double real pole at α when $\lambda = \lambda_{\min}$ and a double real pole at $-\alpha$ when $\lambda = \lambda_{\max}$. It is straightforward to show that this leads to the choices

$$\alpha = \sqrt{p} = \frac{1-\sqrt{\lambda_r}}{1+\sqrt{\lambda_r}} \tag{10}$$

$$\gamma = \frac{(1-\alpha)^2}{\lambda_{\min}} = \frac{(1+\alpha)^2}{\lambda_{\max}}\,, \tag{11}$$

where $\lambda_r = \lambda_{\min}/\lambda_{\max}$. Here α is the maximum modulus of the resulting closed-loop poles (excluding the pole at $z = 1$), and as in (6) it is a function of λ_r. In Fig. 6 we plot these two functions (6) and (10), and we conclude that the estimator in Fig. 4 converges considerably more quickly than the estimator in Fig. 2. This is consistent with the fast static consensus results in (Ghosh et al., 1998; Kokiopoulou and Frossard, 2009; Oreshkin et al., 2010; Liu and Morse, 2011; Erseghe et al., 2011), which show how additional estimator dynamics can significantly improve convergence rates.

There is second problem with the estimator in Fig. 3, one it shares with the estimator in Fig. 4: they are not bounded-input, bounded-state stable. For example, if the estimator outputs y are constant (as they will be in steady state for constant inputs), then the integrator states grow as ramps and are thus unbounded in forward time. To fix this problem, we propose to change the state space on each node from the plane to the cylinder, and in doing so introduce nonlinearities into the dynamics.

We first observe that in the state-space representation (9) of the estimator in Fig. 4, each node i maintains an internal state vector $[x_{ai} \;\; x_{bi}]^T \in \mathbb{R}^2$. As it is the integrator state x_{ai} which can grow unbounded in forward time under bounded inputs, we propose to have it take values on the compact manifold S^1 instead of the real line. We must modify the dynamics to make this work, but the result will be that x_{ai} is automatically bounded regardless of the other signals in the system. In essence, we are proposing dynamic consensus via filtered oscillators, an extension of the unfiltered static oscillator consensus method introduced by Khan (2009).

For convenience we will parameterize the circle S^1 using radian angles in the interval $(-\pi, \pi]$. Also, we will use $\oplus$ and $\ominus$ to denote angle addition, subtraction, and unary negation, so that $\pi \oplus \pi = 0$ and $0 \ominus \pi = \ominus\pi = \pi$. We introduce two functions, the first from $\mathbb{R}$ to S^1 and the second from S^1 to $\mathbb{R}$. The first function is just the standard covering map $\mathcal{C} : \mathbb{R} \to S^1$ which takes any real number s to its projection $\mathcal{C}(s) = s + 2\pi\ell$ onto the circle S^1, where ℓ is the unique integer such that $\mathcal{C}(s) \in (-\pi, \pi]$. The second function is a C^1 phase coupling function $f : S^1 \to \mathbb{R}$. We assume that f is odd, meaning $f(\ominus\theta) = -f(\theta)$ for all $\theta \in S^1$, and that $f(0) = 0$ and $f'(0) = 1$. Otherwise f is free for us to choose, a simple choice being $f = \sin$. Given a constant scaling parameter $\eta > 0$, the new nonlinear estimator for each node i takes the form

$$x_{ai}(k+1) = \tau_i(k) \tag{12}$$

$$x_{bi}(k+1) = p x_{bi}(k) - \gamma\eta \sum_{\{i,j\}=\mathfrak{e}_\ell} w_\ell f\big(\tau_i(k) \ominus \tau_j(k)\big) \tag{13}$$

$$\tau_i(k) = x_{ai}(k) \oplus \mathcal{C}\Big(\frac{x_{bi}(k) + u_i(k)}{\eta}\Big) \tag{14}$$

$$y_i(k) = x_{bi}(k) + u_i(k)\,, \tag{15}$$

where $\tau_i(\cdot)$ is an auxiliary output which takes values in S^1 and is transmitted to neighboring nodes. In what follows, we interpret $\mathcal{C}$, f, and $\oplus$ as acting element-wise on vector arguments to produce vector values. We can thus define the nonlinear Laplacian $\mathcal{L} : (S^1)^n \to \mathbb{R}^n$ as

$$\mathcal{L}(\cdot) = BWf(B^T\cdot), \tag{16}$$

where here we interpret B^T as a $\mathbb{Z}$-linear map from $(S^1)^n$ to $(S^1)^m$. We now write the collection of these estimators (12)–(15) in a more compact form as

$$x_a(k+1) = \tau(k) \tag{17}$$

$$x_b(k+1) = p x_b(k) - \gamma\eta\,\mathcal{L}\big(\tau(k)\big) \tag{18}$$

$$\tau(k) = x_a(k) \oplus \mathcal{C}\Big(\frac{x_b(k) + u(k)}{\eta}\Big) \tag{19}$$

$$y(k) = x_b(k) + u(k)\,, \tag{20}$$

where $\tau = [\tau_1 \;\ldots\; \tau_n]^T$ is a vector in $(S^1)^n$ (as is x_a). Fig. 7 shows a block diagram of this new nonlinear estimator, and we see that it is similar to the one in Fig. 4—we have

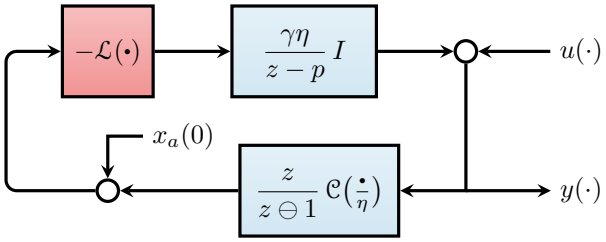

Fig. 7. A nonlinear dynamic average consensus estimator.

merely replaced the Laplacian and integrator blocks with nonlinear versions and introduced the scaling parameter η.

We need the scaling parameter η here because the nonlinear Laplacian $\mathcal{L}(\cdot)$ has a bounded image in $\mathbb{R}^n$. Indeed, the continuous function $|f|$ achieves a maximum value $f_{\max}$ on the compact set S^1, so the Laplacian is bounded as

$$\|\mathcal{L}(\cdot)\|_\infty \leqslant f_{\max}\|BW\|_\infty\,. \tag{21}$$

If we define the error vector $e(k)$ at time k as

$$e(k) = y(k) - u_{\text{ave}}(k)\mathbf{1}\,, \tag{22}$$

then (20) implies

$$x_b(k) = e(k) - \big[u(k) - u_{\text{ave}}(k)\mathbf{1}\big]\,. \tag{23}$$

On the other hand, the ℓ_1-norm of the upper filter in Fig. 7 is $\gamma\eta/(1-p)$, which implies

$$\|x_b\|_\infty \leqslant \frac{\gamma\eta}{1-p}\|\mathcal{L}(\cdot)\|_\infty = \frac{\eta}{\sqrt{\lambda_{\min}\lambda_{\max}}}\|\mathcal{L}(\cdot)\|_\infty\,. \tag{24}$$

Combining (21), (23), and (24), we see that

$$\big\|u - u_{\text{ave}}\mathbf{1}\big\|_\infty \leqslant \|e\|_\infty + \frac{\eta f_{\max}}{\sqrt{\lambda_{\min}\lambda_{\max}}}\|BW\|_\infty\,. \tag{25}$$

For the weights in (2), one can show that $\|BW\|_\infty < 1$ for any graph, which gives us the graph-independent bound

$$\big\|u - u_{\text{ave}}\mathbf{1}\big\|_\infty \leqslant \|e\|_\infty + \frac{\eta f_{\max}}{\sqrt{\lambda_{\min}\lambda_{\max}}}\,. \tag{26}$$

We can view this inequality as a necessary condition on the scaling parameter η. However, choosing η to satisfy (26) for a desired bound on the steady-state error requires knowledge of an upper bound on the differences between the inputs u_i and their global average u_{ave}. Hence this estimator cannot achieve bounded error for general ramp inputs or other types of unbounded input signals, even when the estimator is cascaded, because eventually the bounded nonlinear Laplacian will be unable to produce large enough outputs. This is a clear disadvantage of this nonlinear estimator, but in practice it may not matter because the input signals in many applications will be bounded.

For the remainder of this paper, we will assume for simplicity that there exists $b \in (0,\pi)$ such that $f(\theta) = \theta$ when $|\theta| \leqslant b$. This means that the nonlinear estimator in Fig. 7 will generate the same output as the linear estimator in Fig. 4 when all initial states are zero and the scaling parameter η is sufficiently large relative to the size of the inputs. In this case, if u is constant and η satisfies

$$\big\|u - u_{\text{ave}}\mathbf{1}\big\|_\infty \leqslant \frac{\eta b\sqrt{\lambda_r}}{2\sqrt{n}}\,, \tag{27}$$

then there is an open set of initial node states from which the estimator outputs $y_i(k)$ all converge to the global average u_{ave} as $k \to \infty$. Furthermore, the convergence is exponential with a rate determined by α in (10) (thus it is as fast as the estimator in Fig. 4). Finally, simulations suggest that this open set of initial node states (namely, the estimator's region of attraction) is large for sufficiently large values of η.

For time-varying inputs, we can examine the Bode plot of the linear estimator in Fig. 4 to determine the response of the nonlinear estimator in Fig. 7, assuming η is sufficiently large relative to $\big\|u - u_{\text{ave}}\mathbf{1}\big\|_\infty$. If we exclude the consensus direction in which the estimator acts like an all-pass filter, then it is a high-pass filter with a magnitude of zero at zero frequency increasing to a magnitude of one at a critical frequency θ_c. If θ_c is measured in units of radians per sample and if the Laplacian L has nonzero eigenvalues in the interval $[\lambda_{\min}, \lambda_{\max}]$, then

$$\theta_c \geqslant \cos^{-1}\Big(\frac{1}{2} + \frac{\alpha}{\alpha^2+1}\Big)\,, \tag{28}$$

with equality when $\lambda_{\min}$ is actually an eigenvalue of L. Input frequencies below θ_c will be attenuated in the disagreement directions (which is what we want for consensus), whereas inputs frequencies above θ_c will be amplified. Thus to achieve a small steady-state error with this estimator, the input spectrum should have all significant components below θ_c. Note that cascading the estimator will add the Bode plots and thus further attenuate frequencies below θ_c, but each additional cascade stage requires an additional communicated real value per node per time step. Moreover, as cascading will further amplify frequencies above θ_c, noise and quantization effects will limit the number of useful cascade stages.

4. SIMULATIONS

To test the estimators in Figs. 2 and 7, we simulated their responses on an Erdős-Rényi random graph having $n = 100$ nodes and an edge probability of 0.05. We used the weights in (2) with the hard bound $\lambda_{\max} = 1$, and we chose $\lambda_{\min} = 0.03$ as random trials indicate that this value is a lower bound on all non-zero Laplacian eigenvalues for over 99% of such random graphs. We then chose our estimator parameters according to (5), (10), and (11). The actual minimum and maximum eigenvalues for our random L were 0.05 and 0.9, respectively. Finally, we chose the phase coupling function f so that $f(\theta) = \theta$ for $|\theta| \leqslant 2.5$.

To choose a value for the scaling parameter η, we assumed that our inputs satisfied $\big\|u - u_{\text{ave}}\mathbf{1}\big\|_\infty \leqslant 10$. In this case the necessary condition (26) with $e = 0$ yielded $\eta \geqslant 0.7$, whereas the sufficient condition (27) yielded $\eta \geqslant 462$. We chose $\eta = 500$ for our simulations, although much smaller values seemed to work as well.

We first simulated the estimators in Figs. 2 and 7 with constant inputs chosen uniformly at random over the interval $[0, 10]$ and all initial states set to zero. The results are shown in Fig. 8 under "zero IC." As expected, both estimators generate errors that converge exponentially to zero, but the estimator in Fig. 7 converges considerably more quickly. We then chose initial states uniformly at random over the interval $[-10, 10]$, with the exception of the initial oscillator states for the nonlinear estimator in Fig. 7, which we chose at random from the smaller interval $[-10/\eta, 10/\eta]$ to account for the scaling effects of the parameter η. The results are shown in Fig. 8 under

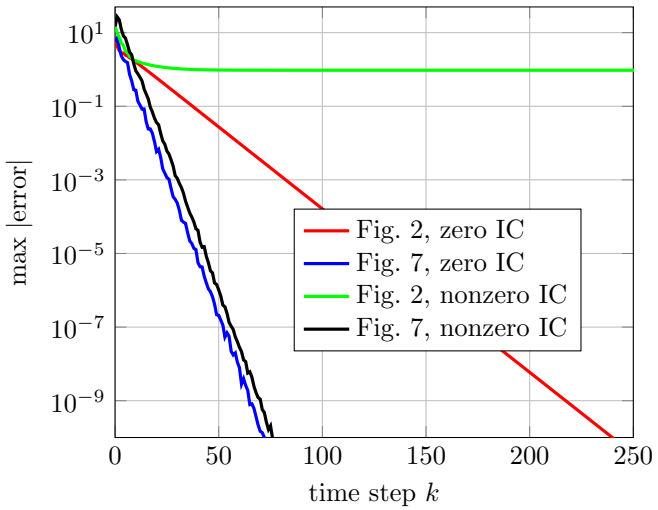

Fig. 8. Maximum errors of the estimators in Figs. 2 and 7 under constant inputs.

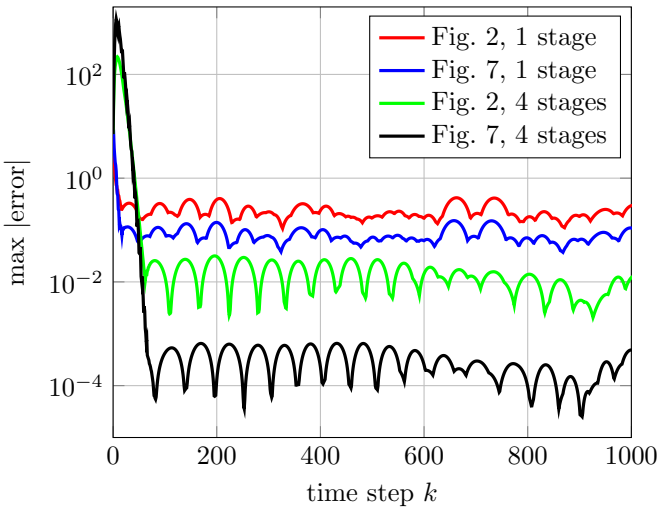

Fig. 9. Maximum errors of the estimators in Figs. 2 and 7 under filtered noise inputs and zero initial states.

"nonzero IC,", and we see that only the estimator in Fig. 7 demonstrates the initialization robustness property.

We then simulated the estimators over time-varying inputs generated as follows. For each node i, we chose a mean μ_i and standard deviation σ_i independently at random from a uniform distribution over the interval $[0, 10]$. Then we created an i.i.d. sequence $v_i(k) \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $k = 0, 1, \ldots$. Finally, we passed the sequence v_i through a seventh-order elliptic filter with a cutoff frequency of 0.02π radians per sample to obtain the input signal u_i for node i. We chose this cutoff frequency to be below the critical frequency θ_c in (28), which for our parameters gave $\theta_c = 0.077\pi$. The results are shown in Fig. 9 for the case of zero initial states, both for a single estimator and for a cascade of four estimators (in the cascade, the inputs enter the first stage, the outputs of the first stage are the inputs to the second, etc., and the outputs of the last stage are the estimates). We see that cascading the estimators reduced the steady-state errors but worsens the transient part of the response. Also, the four-stage estimator requires four times as much communication per node per time step as the single-stage estimator. For nonzero initial states, the results (not shown) are similar for the estimator in Fig. 7, but the steady-state errors are larger for the estimator in Fig. 2 due to its lack of initialization robustness.

REFERENCES

Aragüés, R., Cortés, J., and Sagüés, C. (2012). Distributed consensus on robot networks for dynamically merging feature-based maps. *IEEE Transactions on Robotics*, 28(4), 840–854.

Bai, H., Freeman, R.A., and Lynch, K.M. (2010). Robust dynamic average consensus of time-varying inputs. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 3104–3109. Atlanta, Georgia.

Bai, H., Freeman, R.A., and Lynch, K.M. (2011). Distributed Kalman filtering using the internal model average consensus estimator. In *Proceedings of the 2011 American Control Conference*, 1500–1505. San Francisco, California.

Cortés, J. (2009). Distributed kriged Kalman filter for spatial estimation. *IEEE Trans. Automat. Contr.*, 54(12), 2816–2827.

Elwin, M.L., Freeman, R.A., and Lynch, K.M. (2013). A systematic design process for internal model average consensus estimators. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, 5878–5883. Florence, Italy.

Elwin, M.L., Freeman, R.A., and Lynch, K.M. (2014). Worst-case optimal average consensus estimators for robot swarms. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 3814–3819. Chicago, Illinois.

Erseghe, T., Zennaro, D., Dall'Anese, E., and Vangelista, L. (2011). Fast consensus by the alternating direction multipliers method. *IEEE Trans. Signal Process.*, 59(11), 5523–5537.

Freeman, R.A., Nelson, T.R., and Lynch, K.M. (2010). A complete characterization of a class of robust linear average consensus protocols. In *Proceedings of the 2010 American Control Conference*, 3198–3203. Baltimore, Maryland.

Ghosh, B., Muthukrishnan, S., and Schultz, M.H. (1998). First- and second-order diffusive methods for rapid, coarse, distributed load balancing. *Theory of Computing Systems*, 31, 331–354.

Khan, U.A. (2009). *High Dimensional Consensus in Large-Scale Networks: Theory and Applications*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Kokiopoulou, E. and Frossard, P. (2009). Polynomial filtering for fast convergence in distributed consensus. *IEEE Trans. Signal Process.*, 57(1), 342–354.

Liu, J. and Morse, A.S. (2011). Accelerated linear iterations for distributed averaging. *Annual Reviews in Control*, 35, 160–165.

Lynch, K.M., Schwartz, I.B., Yang, P., and Freeman, R.A. (2008). Decentralized environmental modeling by mobile sensor networks. *IEEE Trans. Robot.*, 24(3), 710–724.

Moallemi, C.C. and Van Roy, B. (2006). Consensus propagation. *IEEE Transactions on Information Theory*, 52(11), 4753–4766.

Montijano, E., Montijano, J.I., Sagüés, C., and Martínez, S. (2014a). Robust discrete time dynamic average consensus. *Automatica*, 50(12), 3131–3138.

Montijano, E., Montijano, J.I., Sagüés, C., and Martínez, S. (2014b). Step size analysis in discrete-time dynamic average consensus. In *Proceedings of the 2014 American Control Conference*, 5127–5132. Portland, Oregon.

Olfati-Saber, R. (2005). Distributed Kalman filter with embedded consensus filters. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, 8179–8184. Seville, Spain.

Olfati-Saber, R. (2007). Distributed Kalman filtering for sensor networks. In *Proceedings of the 46th IEEE Conference on Decision and Control*, 5492–5498. New Orleans. doi:10.1109/CDC.2007.4434303.

Olfati-Saber, R. and Shamma, J.S. (2005). Consensus filters for sensor networks and distributed sensor fusion. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, 6698–6703. Seville, Spain.

Oreshkin, B.N., Coates, M.J., and Rabbat, M.G. (2010). Optimization and analysis of distributed averaging with short node memory. *IEEE Trans. Signal Process.*, 58(5), 2850–2865.

Peterson, C.K. and Paley, D.A. (2013). Distributed estimation for motion coordination in an unknown spatially varying flowfield. *Journal of Guidance, Control, and Dynamics*, 36(3), 894–898. doi:10.2514/1.59453.

Spanos, D.P., Olfati-Saber, R., and Murray, R.M. (2005). Dynamic consensus for mobile networks. In *Proceedings of the 2005 IFAC World Congress*. Prague. Preprint.

Tsitsiklis, J.N. (1984). *Problems in Decentalized Decision-Making and Computation*. Ph.D. thesis, MIT, Cambridge, MA.

Van Scoy, B., Freeman, R.A., and Lynch, K.M. (2014). Asymptotic mean ergodicity of average consensus estimators. In *Proceedings of the 2014 American Control Conference*, 4696–4701. Portland, Oregon.

Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging. *Syst. Contr. Lett.*, 53, 65–78.

Yang, P., Freeman, R.A., and Lynch, K.M. (2008). Multi-agent coordination by decentralized estimation and control. *IEEE Trans. Automat. Contr.*, 53(11), 2480–2496.

Yuan, Y., Liu, J., Murray, R.M., and Gonçalves, J. (2012). Decentralised minimal-time dynamic consensus. In *Proceedings of the 2012 American Control Conference*, 800–805. Montreal.

Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, 46(2), 322–329.