IEEE TRANSACTIONS ON AUTOMATIC CONTROL, VOL. 64, NO. 11, NOVEMBER 2019


# Robust Dynamic Average Consensus Algorithms

Jemin George and Randy A. Freeman

***Abstract*—This technical note considers the dynamic average consensus problem, where a group of networked agents are required to estimate the average of their time-varying reference signals. Almost all existing solutions to this problem require a specific initialization of the estimator states, and such constraints render the algorithms vulnerable to network disruptions. Here, we present three robust algorithms that do not entail any initialization criteria. Furthermore, the proposed algorithms do not rely on the full knowledge of the dynamics generating the reference signals nor assume access to its time derivatives. Two of the proposed algorithms focus on undirected networks and make use of an adaptive scheme that removes the explicit dependence of the algorithm on any upper bounds on the reference signals or its time derivatives. The third algorithm presented here provides a robust solution to the dynamic average consensus problem on directed networks. Compared to the existing algorithms for directed networks, the proposed algorithm guarantees an arbitrarily small steady-state error bound that is independent of any bounds on the reference signals or its time derivatives. The current formulation allows each agent to select its own performance criteria, and the algorithm parameters are distributedly selected such that the most stringent requirement among them is satisfied. A performance comparison of the proposed approach to existing algorithms is presented.**

***Index Terms*—Distributed average tracking, dynamic average consensus, finite-time convergence, initialization error, multi-agent systems, weighted directed graph.**

## I. Introduction

Consider a set of $n$ networked agents, each with its own reference signal $\phi_i(t)$, $i = 1, \ldots, n$. The *dynamic average consensus* (DAC) problem involves designing distributed algorithms that would allow the agents to locally estimate the time-varying average $\bar{\phi}(t) \triangleq \frac{1}{n}\sum_{i=1}^{n}\phi_i(t)$. This estimator design problem has numerous applications in multiagent systems. For example, in the distributed optimization problem, $\min_x \sum_{i=1}^{n} f_i(x)$, $\phi_i(t)$ s are the local gradients $\nabla f_i(\cdot)$; in the distributed estimation problem, $\phi_i(t)$ s are local weighted measurement residuals; and in multiagent coordination problems such as containment control, $\phi_i(t)$ s are the leader trajectories. Thus, DAC is at the heart of numerous network applications, such as distributed learning, distributed sensor fusion, formation control, distributed optimization, and distributed mapping.

The main difficulty in designing distributed solutions to the DAC problem is the lack of access to any error signals. To be more specific, if $x_i(t)$ is the $i$th-node's estimate of $\bar{\phi}(t)$, then none of the nodes have access to the average-consensus error $\tilde{x}_i(t) = x_i(t) - \bar{\phi}(t)$, thus rendering the traditional feedback-control techniques inapplicable. Therefore, solutions to the DAC problem were first proposed for reference signals with steady-state values [1] or slowly varying reference signals [2]. Assuming access to the dynamics that generate the reference signal, internal-model-based DAC algorithms are presented in [3] and [4]. Assuming access to the time derivatives of the reference signals, a DAC algorithm built on singular perturbation theory is given in [5]. In many real-world applications, it is not reasonable to assume knowledge of the reference signal dynamics or presume access to its time derivatives.

Even though the agents do not have direct access to the error signal $\tilde{x}_i(t)$, they can calculate the local difference or disagreement in the error, i.e., $\tilde{x}_i(t) - \tilde{x}_j(t) = x_i(t) - x_j(t)$. Thus, the DAC problem is solved if the error signal is such that it sums to zero, and the agents reach consensus on $\tilde{x}_i$. In other words, if $\sum_{i=1}^{n}\tilde{x}_i(t) = 0$ and $\tilde{x}_i(t) = \tilde{x}_j(t)$ for all $(i, j)$ pairs, then $\tilde{x}_i(t) = 0$ for all $i$. Therefore, there exist several solutions to DAC problems, where an estimator is designed such that the estimator structure, along with an initialization requirement, provides the zero-sum condition $\sum_{i=1}^{n}\tilde{x}_i(t) = 0$, while the inputs to the estimator are selected to guarantee that the agents reach consensus on the error signal. Examples of such algorithms include the nonlinear DAC estimators for reference signals with bounded derivatives given in [6]–[10]. The algorithms in [6]–[10] are shown to yield bounded average-consensus error even for a directed network, but the error bounds are proportional to the upper bound on the time derivatives of the reference signals. Currently, there exists no DAC algorithm for directed networks that can yield arbitrarily small average-consensus error, independent of the reference signal bounds, without assuming known reference signal dynamics or requiring access to the time derivatives of the reference signals.

While the DAC problem focuses on designing estimators, a combined estimator and controller design problem to estimate and track the time-varying average signal has been studied under the name *distributed average tracking* (DAT) [11]–[20]. The problem formulation in DAT consists of assuming a particular dynamic model for individual agents and then designing a distributed input signal that allows the agents to track the time-varying average of individual reference signals. The main drawback to considering a combined estimator/controller solution is that it is tailored to specific node dynamics and, therefore, only valid for the assumed dynamic model. As a result, there exist numerous DAT solutions to the same average-consensus problem involving agents with single-integrator dynamics [11], [12], double-integrator dynamics [15], [17], Euler–Lagrange dynamics [14], [16], known linear dynamics [13], nonlinear dynamics [18], heterogeneous dynamics [20], and so on. Furthermore, the combined estimator/controller solution limits the utility of such algorithms for numerous network applications such as distributed learning, distributed sensor fusion, formation control, and distributed mapping. Besides, if the agents are able to estimate the time-varying average, say using a DAC estimator, then the control design problem is often trivial.

The DAC algorithms in [6]–[10] require a specific initialization of its variables to satisfy the condition $\sum_{i=1}^{n}\tilde{x}_i(t_0) = 0$. This requirement seems benign at first because it can be easily satisfied by

Manuscript received June 12, 2018; revised November 9, 2018; accepted February 9, 2019. Date of publication February 26, 2019; date of current version October 30, 2019. This work was supported in part by the ONR under Grant N00014-16-1-2106. Recommended by Associate Editor G. Gu. *(Corresponding author: Jemin George.)*

J. George is with the U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: jemin.george.civ@mail.mil).

R. A. Freeman is with Northwestern University, Evanston, IL 60208 USA (e-mail: freeman@eecs.northwestern.edu).

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

Digital Object Identifier 10.1109/TAC.2019.2901819

selecting $x_i(t_0) = \phi_i(t_0)$ for all $i$. However, when an agent leaves the network or when the network splits into several small subgraphs, the condition $\sum_{i=1}^{n} \tilde{x}_i(t_0) = 0$ is violated. This results in a nonzero steady-state error unless all the nodes reinitialize the algorithm after every such network disruption. This algorithm sensitivity to initialization, typically referred to as the lack of robustness to *initialization errors*, is an issue in all existing DAC algorithms and in most DAT approaches.

Here, we present three robust solutions to the DAC problem that do not require any initialization criteria. The proposed algorithms do not rely on the full knowledge of the dynamics generating the reference signals nor assume access to its time derivatives. Among the three robust algorithms, the first two tackle the DAC problem over undirected networks, while the third focuses on directed networks. The two robust algorithms for undirected networks make use of an adaptive scheme to guarantee zero steady-state consensus error, while removing the explicit dependence of the algorithm on any upper bounds on the reference signal or its time derivatives. Compared to existing DAC algorithms for directed networks, the third algorithm presented here is robust to initialization errors and guarantees finite-time convergence of the consensus error to an arbitrarily small neighborhood of origin regardless of any bounds on the reference signal or its time derivatives. Furthermore, the proposed DAC algorithm for directed networks allows each agent to select its own performance criteria, and the algorithm parameters are distributedly selected such that the most stringent requirement among them is satisfied. Performance comparison of the proposed approaches to existing algorithms, as well as numerical simulation of the robust algorithm, is also presented.

## II. Preliminaries

### A. Notation

Let $\mathbb{R}^{n\times m}$ denote the set of $n \times m$ real matrices. Boldface lowercase letters are used to represent vectors, while matrices are denoted by capital letters. For a vector $\boldsymbol{\phi}$, $\phi_i$ or $[\boldsymbol{\phi}]_i$ is the $i$th entry of $\boldsymbol{\phi}$. An $n \times n$ identity matrix is denoted as $I_n$ and $\mathbf{1}_n$ denotes an $n$-dimensional vector of all ones. Let $\mathbb{R}^n_{\mathbf{1}}$ denote the set of all $n$-dimensional vectors of the form $\kappa\mathbf{1}_n$, where $\kappa \in \mathbb{R}$. The absolute value of a vector is given as $|\mathbf{x}| = [\,|x_1| \quad \ldots \quad |x_n|\,]^\top$. Let $\operatorname{sgn}\{\cdot\}$ denote the signum function, and $\forall\, \mathbf{x} \in \mathbb{R}^n$, $\operatorname{sgn}\{\mathbf{x}\} \triangleq [\,\operatorname{sgn}\{x_1\} \quad \ldots \quad \operatorname{sgn}\{x_n\}\,]^\top$. For $p \in [1, \infty]$, the $p$-norm of a vector $\mathbf{x}$ is denoted as $\|\mathbf{x}\|_p$.

### B. Network Model

For a weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order $n$, $\mathcal{V} \triangleq \{v_1, \ldots, v_n\}$ represents the agents or nodes, and the communication links between the agents are represented as $\mathcal{E} \triangleq \{e_1, \ldots, e_\ell\} \subseteq \mathcal{V} \times \mathcal{V}$. Let $\mathcal{I}$ denote the index set $\{1, \ldots, n\}$, and $\forall\, i \in \mathcal{I}$, define $\mathcal{N}_i \triangleq \{v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}\}$ to denote the set of incoming neighbors of node $v_i$. Let $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n\times n}$ be the *adjacency matrix* with entries of $a_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$ and zero otherwise. Define $\Delta = \operatorname{diag}(\mathcal{A}\mathbf{1}_n)$ as the *degree matrix* and $\mathcal{L} = \Delta - \mathcal{A}$ as the graph *Laplacian*. The *incidence matrix* of the graph is defined as $\mathcal{B} = [b_{ij}] \in \{-1, 0, 1\}^{n\times \ell}$, where $b_{ij} = -1$ if edge $e_j$ leaves node $v_i$, $b_{ij} = 1$ if edge $e_j$ enters node $v_i$, and $b_{ij} = 0$ otherwise. If $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is undirected, then each undirected edge is considered as two distinct directed edges, and the edges are labeled such that they are grouped into incoming links to nodes $v_1$ to $v_n$.

*Lemma 1:* For any strongly connected weight-balanced graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order $n$, the graph Laplacian $\mathcal{L}$ is a positive-semidefinite matrix with a single eigenvalue at 0 corresponding to both the left and right eigenvectors $\mathbf{1}_n^\top$ and $\mathbf{1}_n$, respectively.

*Proof:* See [21]. ■

It follows from Lemma 1 that $\forall\, \mathbf{x} \in \mathbb{R}^n$, we have, $\mathbf{x}^\top\mathcal{L}\mathbf{x} = \mathbf{x}^\top\mathcal{L}^\top\mathbf{x} \geq 0$, and $\mathbf{x}^\top\mathcal{L}\mathbf{x} = 0$ if and only if $\mathbf{x} \in \mathbb{R}^n_{\mathbf{1}}$.

## III. Problem Formulation

Let $\phi_i(t) \in \mathbb{R}$ denote the $i$th-node's ($v_i$'s) reference signal at time $t$. The DAC problem involves each agent estimating the time-varying signal

$$\bar{\phi}(t) = \frac{1}{n}\sum_{i=1}^{n} \phi_i(t) = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{\phi}(t) \tag{1}$$

where $n$ is the number of agents and $\boldsymbol{\phi}(t) \in \mathbb{R}^n \triangleq [\phi_1(t) \quad \ldots \quad \phi_n(t)]^\top$. Let $\dot{\boldsymbol{\phi}}(t) \in \mathbb{R}^n \triangleq [\,\dot{\phi}_1(t) \quad \ldots \quad \dot{\phi}_n(t)\,]^\top$ denote the time derivative of $\boldsymbol{\phi}(t)$. Now, we make the following standing assumption regarding $\dot{\boldsymbol{\phi}}(t)$.

*Assumption 1:* For any two one-hop neighbors in $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the local difference in $\dot{\phi}_i(t)$ are bounded such that there exists a positive constant $\dot{\varphi}$ that satisfies

$$\sup_{\substack{t\in[t_0,\infty)\\ \forall i,j:(v_i,v_j)\in\mathcal{E}}} \left|\dot{\phi}_i(t) - \dot{\phi}_j(t)\right| \leq \dot{\varphi} < \infty. \tag{2}$$

Note that Assumption 1 is less strict than assuming absolute bounds on signals $\dot{\phi}_i(t)$. Using vector notation, (2) can be written as

$$\sup_{t\in[t_0,\infty)} \left\|\mathcal{B}^\top\dot{\boldsymbol{\phi}}(t)\right\|_\infty \leq \dot{\varphi} \tag{3}$$

where $\mathcal{B}$ is the incidence matrix. Here, scalar reference signals are considered solely for the purpose of simplifying the analysis and notations. The current formulation and results can be easily extended to multidimensional scenarios.

## IV. Robust DAC Algorithms for Undirected Networks

In this section, we discuss two robust DAC algorithms for undirected networks. We make the following assumption regarding the underlying network.

*Assumption 2:* The interaction topology of $n$ networked agents is given as an unweighted[1] connected undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

It follows from Lemma 1 that the network Laplacian $\mathcal{L}$ is a positive-semidefinite matrix with a single eigenvalue at 0 corresponding to both the left and right eigenvectors $\mathbf{1}_n^\top$ and $\mathbf{1}_n$, respectively.

*Lemma 2:* Let $M \triangleq (I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)$. For any connected undirected network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ of order $n$, the graph Laplacian $\mathcal{L}$ and the incidence matrix $\mathcal{B}$ satisfy

$$M = \mathcal{L}(\mathcal{L})^+ = \mathcal{B}\mathcal{B}^\top\left(\mathcal{B}\mathcal{B}^\top\right)^+ = \mathcal{B}\left(\mathcal{B}^\top\mathcal{B}\right)^+\mathcal{B}^\top \tag{4}$$

where $(\cdot)^+$ denotes the generalized inverse.

*Proof:* See [22, Lemma 3]. ■

### A. Nonrobust DAC Algorithm

Consider the following DAC algorithm given in [12]:

$$\dot{z}_i(t) = -\alpha\sum_{j=1}^{n} a_{ij}\operatorname{sgn}\{x_i(t) - x_j(t)\}, \quad z_i(t_0) \tag{5a}$$

$$x_i(t) = z_i(t) + \phi_i(t) \tag{5b}$$

[1] $\mathcal{A} = [a_{ij}] \in \{0, 1\}^{n\times n}$.

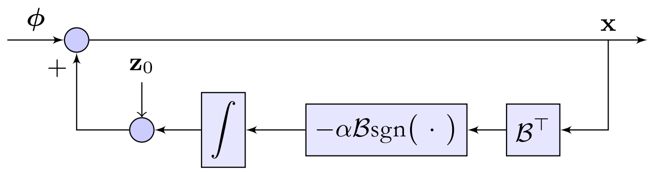



Fig. 1. Nonrobust DAC algorithm.

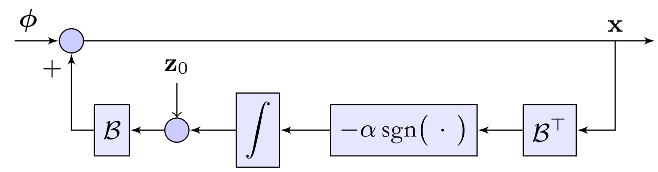



Fig. 2. Finite-time convergent robust DAC algorithm.

where $z_i(t)$ is the internal estimator state for agent $v_i$ and $x_i(t)$ is agent $v_i$'s estimate of $\bar{\phi}(t)$. Using vector notation, (5) can be written as

$$\dot{\mathbf{z}}(t) = -\alpha\mathcal{B}\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\}, \quad \mathbf{z}(t_0) = \mathbf{z}_0 \tag{6a}$$

$$\mathbf{x}(t) = \mathbf{z}(t) + \boldsymbol{\phi}(t) \tag{6b}$$

where $\mathbf{z}(t) \in \mathbb{R}^n$ and $\mathbf{x}(t) \in \mathbb{R}^n$. The objective here is to force $\tilde{\mathbf{x}}(t) \triangleq \mathbf{x}(t) - \bar{\phi}(t)\mathbf{1}_n$ to zero. The finite-time convergence proof of $\tilde{\mathbf{x}}(t) \to \mathbf{0}_n$ given in [12] consists of the following two steps.

*Step 1:* Zero-sum condition:

From (6b), $\tilde{\mathbf{x}}(t) = \mathbf{z}(t) + M\boldsymbol{\phi}(t)$. Thus, $\mathbf{1}_n^\top\tilde{\mathbf{x}}(t) = \mathbf{1}_n^\top\mathbf{z}(t)$. After substituting (6a), $\dot{\tilde{\mathbf{x}}}(t)$ can be written as

$$\dot{\tilde{\mathbf{x}}}(t) = -\alpha\mathcal{B}\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + M\dot{\boldsymbol{\phi}}(t). \tag{7}$$

Thus, $\mathbf{1}_n^\top\dot{\tilde{\mathbf{x}}}(t) = 0$ for all $t \geq t_0$. Therefore, if $\mathbf{1}_n^\top\mathbf{z}(t_0) = 0$, then $\mathbf{1}_n^\top\tilde{\mathbf{x}}(t) = 0$ for all $t \geq t_0$.

*Step 2:* Consensus on error:

Define $\mathbf{y}(t) = \mathcal{B}^\top\tilde{\mathbf{x}}(t) = \mathcal{B}^\top\mathbf{x}(t)$. After substituting (7), $\dot{\mathbf{y}}(t)$ can be written as

$$\dot{\mathbf{y}}(t) = -\alpha\mathcal{B}^\top\mathcal{B}\,\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + \mathcal{B}^\top\dot{\boldsymbol{\phi}}(t) \tag{8}$$

where we used the fact $\mathcal{B}^\top M = \mathcal{B}^\top$. Finite-time convergence of $\mathbf{y}(t)$ to the origin is guaranteed if $\alpha > \sup_{t\geq t_0}\|\dot{\boldsymbol{\phi}}(t)\|_\infty$ (see [12, Th. 1]).

In summary, if (6a) is initialized such that $\mathbf{1}_n^\top\mathbf{z}_0 = 0$ and the gain, $\alpha$, is selected such that $\alpha > \sup_{t\geq t_0}\|\dot{\boldsymbol{\phi}}(t)\|_\infty$, then the DAC algorithm in (6) guarantees finite-time convergence of $\tilde{\mathbf{x}}(t) \to \mathbf{0}_n$.

Although the initialization requirement $\mathbf{1}_n^\top\mathbf{z}_0 = 0$ can be easily satisfied by selecting $z_i(t_0) = 0$ for all $i \in \mathcal{I}$, when an agent leaves the network or when the network divides, the condition $\mathbf{1}_n^\top\mathbf{z}_0 = 0$ is violated. This would result in a nonzero steady-state error unless all the agents reinitialize the algorithm after every such network disruption.

A block diagram of the resulting algorithm is given in Fig. 1. Note that there is a direct feedthrough from $\mathbf{z}_0$ to $\mathbf{x}(t)$ in Fig. 1. If $\mathbf{z}_0$ has nonzero components along the consensus direction, then the estimator input, $-\alpha\mathcal{B}\mathrm{sgn}\{\mathcal{B}^\top\mathbf{x}(t)\}$, will not be able to account for this nonzero component since $\mathcal{B}^\top\mathbf{1}_n = 0$. We now present two novel techniques to resolve the lack of robustness caused by the direct feedthrough from $\mathbf{z}_0$ to $\mathbf{x}(t)$ in Fig. 1.

## *B. Robust DAC Algorithm I*

The first approach to prevent the direct feedthrough from $\mathbf{z}_0$ to $\mathbf{x}(t)$ involves placing a Laplacian or an incidence matrix in the feedback loop such that only the components of $\mathbf{z}(t)$ along the disagreement direction show up in $\mathbf{x}(t)$. This can be easily achieved by removing the incidence matrix $\mathcal{B}$ from the estimator input block and placing it after $\mathbf{z}_0$, as shown in Fig. 2.

Note that displacing the incidence matrix does not affect the condition $\forall t$, $\mathbf{1}_n^\top\tilde{\mathbf{x}}(t) = 0$, and the algorithm retains the same average-consensus error dynamics as the original nonrobust algorithm. Thus, the finite-time convergence of the robust algorithm in Fig. 2 can be proven using the two steps outlined earlier. However, finite-time convergence requires that both the nonrobust and robust algorithms select the control gain $\alpha$ such that $\alpha > \sup_{t\geq t_0}\|\dot{\boldsymbol{\phi}}(t)\|_\infty$. Note that this is a global bound on $\dot{\boldsymbol{\phi}}(t)$, and in many real-world applications, such bounds may not be known to all the agents. Therefore, we further enhance the robust algorithm in Fig. 2 by replacing the constant gain $\alpha$ with an adaptive gain $K(t)$. We also replace the known global upper bound assumption with a weaker existence condition given in Assumption 1.

*Theorem 1:* Given Assumptions 1 and 2, the robust DAC algorithm

$$\dot{\mathbf{z}}(t) = -K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\}, \quad \mathbf{z}(t_0) = \mathbf{z}_0 \tag{9a}$$

$$\mathbf{x}(t) = \mathcal{B}\mathbf{z}(t) + \boldsymbol{\phi}(t) \tag{9b}$$

guarantees that the average-consensus error $\tilde{\mathbf{x}}(t)$ asymptotically decays to zero for any initial condition $\mathbf{z}_0$. Here, $K(t)$ is a diagonal gain matrix of the form $\mathrm{diag}([\,k_1(t) \quad \ldots \quad k_\ell(t)\,])$, and the gains $k_l(t)$, $\forall l \in \{1,\ldots,\ell\}$ are updated using the adaptive law

$$\dot{k}_l(t) = |y_l(t)|, \quad k_l(t_0) \geq 1 \tag{10}$$

where $\mathbf{y}(t) \triangleq \left[\,y_1(t)\ \ldots\ y_\ell(t)\,\right]^\top = \mathcal{B}^\top\mathbf{x}(t)$.

*Proof:* We follow the same two-step process outlined earlier. Since the consensus error can be written as $\tilde{\mathbf{x}}(t) = \mathcal{B}\mathbf{z}(t) + M\boldsymbol{\phi}(t)$, we have $\mathbf{1}_n^\top\tilde{\mathbf{x}}(t) = 0$ for all $t \geq t_0$, and this completes the first step.

The second step involved showing $\mathbf{y}(t)$ asymptotically decays to zero, i.e., the agents reach consensus on $\tilde{\mathbf{x}}(t)$. The consensus error dynamics can be written as

$$\dot{\tilde{\mathbf{x}}}(t) = -\mathcal{B}K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + M\dot{\boldsymbol{\phi}}(t). \tag{11}$$

Thus, we have

$$\dot{\mathbf{y}}(t) = -\mathcal{B}^\top\mathcal{B}K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + \mathcal{B}^\top M\dot{\boldsymbol{\phi}}(t).$$

Now, consider a nonnegative function of the form

$$V = \mathbf{y}^\top(t)\left(\mathcal{B}^\top\mathcal{B}\right)^{+\mathbf{y}}(t) + \frac{1}{2}\sum_{l=1}^{\ell}\left(k_l(t) - k^*\right)^2$$

where $k_l(t)$ are the individual control gains and $k^*$ is a constant to be determined. Now, we have

$$\begin{aligned}\dot{V} = &-2\mathbf{y}^\top(t)K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + \sum_{l=1}^{\ell}\tilde{k}_l(t)\dot{k}_l(t)\\ &+ 2\mathbf{y}^\top(t)\left(\mathcal{B}^\top\mathcal{B}\right)^+\mathcal{B}^\top M\dot{\boldsymbol{\phi}}(t)\end{aligned} \tag{12}$$

where $\tilde{k}_l(t) = k_l(t) - k^*$. Note that

$$\left(\mathcal{B}^\top\mathcal{B}\right)^+\mathcal{B}^\top M = \left(\mathcal{B}^\top\mathcal{B}\right)^+\mathcal{B}^\top$$

and $\mathbf{y}^\top(t)K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} = \sum_{l=1}^{\ell}k_l(t)\,|y_l(t)|$. Now, substituting (10) into (12) yields

$$\dot{V} = -\sum_{l=1}^{\ell}k_l(t)\,|y_l(t)| - k^*\sum_{l=1}^{\ell}|y_l(t)| + 2\mathbf{y}^\top(t)\left(\mathcal{B}^\top\mathcal{B}\right)^+\mathcal{B}^\top\dot{\boldsymbol{\phi}}(t).$$

Note that $\mathbf{y}^\top(t)(\mathcal{B}^\top\mathcal{B})^+\mathcal{B}^\top\dot{\boldsymbol{\phi}}(t) \le \|\mathbf{y}(t)\|_1 \|(\mathcal{B}^\top\mathcal{B})^+\|_\infty \times \|\mathcal{B}^\top\dot{\boldsymbol{\phi}}(t)\|_\infty$. Thus, we have

$$\dot{V} \le -\sum_{l=1}^{\ell} k_l(t)\,|y_l(t)| + \left(2\dot{\varphi}\left\|\left(\mathcal{B}^\top\mathcal{B}\right)^+\right\|_\infty - k^*\right)\|\mathbf{y}(t)\|_1 .$$

If $k^*$ is selected such that $k^* \ge 2\dot{\varphi}\|(\mathcal{B}^\top\mathcal{B})^+\|_\infty$, we have $\dot{V} \le -\|\mathbf{y}(t)\|_1$. Thus, $V$ is upper bounded, and therefore, $\mathbf{y}(t)$ and $K(t)$ are bounded. Based on Assumption 1, boundedness of $\mathbf{y}(t)$ and $K(t)$ implies bounded $\dot{\mathbf{y}}(t)$. Since $V$ is lower bounded at zero and $\dot{V} \le -\|\mathbf{y}(t)\|_1 \le -\|\mathbf{y}(t)\|_2$, we have $\int_{t_0}^{\infty}(\mathbf{y}^\top(t)\mathbf{y}(t))^{1/2}dt < \infty$, i.e., $\mathbf{y}(t)$ is square-integrable. Now, based on the BarBălat's Lemma (see [23, Lemma 3.2.5]), we have $\lim_{t\to\infty}\mathbf{y}(t) = \mathbf{0}_\ell$. This completes the second step.

Since $\mathbf{1}_n^\top\tilde{\mathbf{x}}(t) = 0$ for all $t \ge t_0$, $\lim_{t\to\infty}\mathbf{y}(t) = \mathbf{0}_\ell$ implies $\lim_{t\to\infty}\tilde{\mathbf{x}}(t) = \mathbf{0}_n$. ■

Solutions to

$$\dot{\mathbf{y}}(t) = -\mathcal{B}^\top\mathcal{B}K(t)\mathrm{sgn}\left\{\mathbf{y}(t)\right\} + \mathcal{B}^\top M\dot{\boldsymbol{\phi}}(t) \tag{13}$$

are understood in the Filippov sense [24]. Define a vector field $\mathfrak{f}(t,\mathbf{y}(t)) : \mathbb{R}\times\mathbb{R}^\ell \mapsto \mathbb{R}^\ell \triangleq -\mathcal{B}^\top\mathcal{B}K(t)\mathrm{sgn}\left\{\mathbf{y}(t)\right\} + \mathcal{B}^\top M\dot{\boldsymbol{\phi}}(t)$. Note that the Filippov set-valued map for the vector field $\mathfrak{f}(t,\mathbf{y}(t))$ is multiple-valued only at the point of discontinuity, i.e., at the origin. Therefore, the aforementioned stability analysis using a smooth Lyapunov function is valid because the function $V$ is decreasing along every Filippov solution of (13) that starts on $\mathbb{R}^\ell\backslash\{\mathbf{0}\}$. Thus, $\mathbf{y}(t)$ is globally asymptotically stable.

Note that the original algorithm in Fig. 1 is a "node-based" solution, while the robust algorithm in Fig. 2 is an "edge-based" solution. In other words, there is one internal state per node for the original algorithm, while there are multiple internal states per node for the robust algorithm. The number of internal states per node depends on the node degree.

### C. Robust DAC Algorithm II

The second approach to robustifying the DAC algorithm is to make sure that any contribution of $\mathbf{z}_0$ to $\mathbf{x}(t)$ will diminish exponentially fast. This can be achieved by introducing $-\gamma\mathbf{z}(t)$ into (6a), where $\gamma > 0$ is a design parameter. Thus, we have

$$\dot{\mathbf{z}}(t) = -\gamma\mathbf{z}(t) - \mathcal{B}K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\}, \quad \mathbf{z}(t_0) = \mathbf{z}_0 \tag{14a}$$

$$\mathbf{x}(t) = \mathbf{z}(t) + \boldsymbol{\phi}(t). \tag{14b}$$

Here, we replaced the constant gain $\alpha$ with an adaptive gain, $K(t)$, that needs to be determined. Now, the average-consensus error dynamics can be written as

$$\dot{\tilde{\mathbf{x}}}(t) = -\gamma\mathbf{z}(t) - \mathcal{B}K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + M\dot{\boldsymbol{\phi}}(t).$$

Adding and subtracting $\gamma M\boldsymbol{\phi}(t)$ yields

$$\dot{\tilde{\mathbf{x}}}(t) = -\gamma\tilde{\mathbf{x}}(t) - \mathcal{B}K(t)\,\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + M\left(\dot{\boldsymbol{\phi}}(t) + \gamma\boldsymbol{\phi}(t)\right). \tag{15}$$

Compared to the error dynamics given in (11), both $\boldsymbol{\phi}(t)$ and $\dot{\boldsymbol{\phi}}(t)$ are present in the current average-consensus error dynamics. Therefore, we need the following additional assumption on $\boldsymbol{\phi}(t)$ besides the one in Assumption 1.

*Assumption 3:* For any two one-hop neighbors in $\mathcal{G}(\mathcal{V},\mathcal{E})$, the local difference in $\phi_i(t)$ is bounded such that there exists a positive constant $\varphi$ that satisfies

$$\sup_{\substack{t\in[t_0,\infty)\\ \forall i,j:(v_i,v_j)\in\mathcal{E}}} |\phi_i(t) - \phi_j(t)| \le \varphi < \infty. \tag{16}$$

In vector form, (16) can be written as $\sup_{t\in[t_0,\infty)}\|\mathcal{B}^\top\boldsymbol{\phi}(t)\|_\infty \le \varphi$.

*Theorem 2:* Given Assumptions 1–3, the robust DAC algorithm in (14) with the diagonal gain matrix $K(t)$ updated using the adaptive law

$$\dot{k}_l(t) = |y_l(t)|, \quad k_l(t_0) \ge 1 \quad \forall\, l \in \{1,\ldots,\ell\} \tag{17}$$

guarantees that the average-consensus error, $\tilde{\mathbf{x}}(t)$, asymptotically decays to zero for any initial condition $\mathbf{z}_0$. Here, $\mathbf{y}(t) \triangleq [\,y_1(t) \quad \ldots \quad y_\ell(t)\,]^\top = \mathcal{B}^\top\mathbf{x}(t)$.

*Proof:* From (15), we have $\mathbf{1}_n^\top\dot{\tilde{\mathbf{x}}}(t) = -\gamma\mathbf{1}_n^\top\tilde{\mathbf{x}}(t)$. Thus, regardless of the initial condition $\mathbf{z}(t_0)$, $\sum_{i=1}^n \tilde{x}_i(t)$ exponentially decays to 0. After left multiplying (15) with $\mathcal{B}^\top$, we have

$$\dot{\mathbf{y}}(t) = -\gamma\mathbf{y}(t) - \mathcal{B}^\top\mathcal{B}K(t)\mathrm{sgn}\left\{\mathcal{B}^\top\mathbf{x}(t)\right\} + \mathcal{B}^\top M\left(\dot{\boldsymbol{\phi}}(t) + \gamma\boldsymbol{\phi}(t)\right).$$

The global asymptotic stability of $\tilde{\mathbf{x}}(t)$ can be shown through the same procedure given in the proof of Theorem 1 and thus is omitted. ■

*Remark 1:* Due to the nature of the adaptive law, the control gain can only increase. Therefore, persistent disturbances that create transients will cause $K(t)$ to drift higher and higher with time. To prevent this drift, one would have to turn OFF the adaptation when $K(t)$ gets to some threshold value, but to know this threshold value, one needs to know an upper bound on the reference signal and/or its derivative. A similar issue exists even if one were to use the projection-operator-based adaptive law [25].

The main difference between the robust DAC algorithms given in Sections IV-B and IV-C is that the algorithm in Section IV-B is an edge-based algorithm, while the algorithm in Section IV-C is a node-based algorithm. Moreover, the robust algorithm in Section IV-C requires an extra assumption (see Assumption 3) compared to the algorithm in Section IV-B. Both algorithms require the same number of adaptive gains, and the adaptive law is identical in both algorithms. Furthermore, both algorithms guarantee asymptotic stability of the DAC error.

## V. Robust DAC Algorithm for Directed Networks

In this section, we present the formulation and analysis of a robust DAC algorithm for directed networks. It is imperative to recognize that there is no trivial extension of the robust algorithms given in Sections IV-B and IV-C to the direct network case because the algorithms given in Sections IV-B and IV-C are built on the property $\mathcal{L} = \mathcal{B}\mathcal{B}^\top$, which is unique to undirected (or symmetric) graphs. For directed graphs, $\mathcal{L} \ne \mathcal{B}\mathcal{B}^\top$, and the agents have no distributed way of calculating quantities like $\mathcal{B}\mathcal{B}^\top\mathbf{x}(t)$.

### A. Preliminaries

We make the following assumption regarding the underlying network:

*Assumption 4:* The interaction topology of $n$ networked agents is given as a strongly connected weight-balanced digraph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

*Lemma 3:* For any strongly connected weight-balanced digraph of order $n$, we have $M \triangleq (I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top) = \mathcal{L}(\mathcal{L})^+ = (\mathcal{L})^+\mathcal{L}$, where $(\cdot)^+$ denotes the generalized inverse.

*Proof:* Consider the singular value decomposition $\mathcal{L} = U\Sigma V^\top$. Thus, $(\mathcal{L})^+ = V\Sigma^+U^\top$. Note that $U$ and $V$ are orthogonal matrices with $\frac{1}{\sqrt{n}}\mathbf{1}_n$ as the last column vector corresponding to the singular value 0. Therefore, we have

$$\mathcal{L}(\mathcal{L})^+ = U\Sigma V^\top V\Sigma^+U^\top = U\Sigma\Sigma^+U^\top.$$

Since $\Sigma\Sigma^+ = \left[\begin{smallmatrix} I_{n-1} & \\ 0 & 0\end{smallmatrix}\right]$, we have

$$\mathcal{L}\mathcal{L}^+ = \left[\,\mathbf{u}_1 \quad \ldots \quad \mathbf{u}_{n-1} \quad \tfrac{1}{\sqrt{n}}\mathbf{1}_n\,\right]\left[\,\mathbf{u}_1 \quad \ldots \quad \mathbf{u}_{n-1} \quad \mathbf{0}_n\,\right]^\top$$

where $\mathbf{u}_i$ are the column vectors of $U$. Therefore, all the diagonal elements of $\mathcal{L}\mathcal{L}^+$ are $1 - \frac{1}{n}$, and all the off-diagonal entries are $-\frac{1}{n}$. Similarly, we have

$$(\mathcal{L})^+ \mathcal{L} = V\Sigma^+ U^\top U \Sigma V^\top = V\Sigma^+ \Sigma V^\top$$

and the above argument holds. ∎

The robust DAC algorithm given in this section requires the following assumption regarding $\dot{\boldsymbol{\phi}}(t)$.

*Assumption 5:* Signals $\dot{\phi}_i(t)$ are bounded such that $\forall\, i \in \mathcal{I}$, there exist known local bounds $\vartheta_i > 0$ that satisfy

$$\sup_{t\in[t_0,\infty)} \left|\dot{\phi}_i(t)\right| \leq \vartheta_i < \infty. \tag{18}$$

Since the bounds given in Assumption 5 are local bounds on the reference signals, the robust DAC algorithm proposed in this section requires the agents to reach consensus on the maximum of these bounds. There exist several different max-consensus algorithms that guarantee that the agents reach consensus on the maximum in finite time [26]–[28]. Here, we implement the following algorithms:

$$\dot{\hat{\vartheta}}_i(t) = \mathrm{sgn}_+ \left(\sum_{j\in\mathcal{N}_i} a_{ij}\left(\hat{\vartheta}_j(t) - \hat{\vartheta}_i(t)\right)\right), \ \hat{\vartheta}_i(t_0) = \vartheta_i \tag{19}$$

where $\mathrm{sgn}_+ : \mathbb{R} \mapsto \mathbb{R}$ is defined as $\mathrm{sgn}_+(x) = 1$ if $x > 0$ and zero otherwise. Here, $\hat{\vartheta}_i(t)$ denotes the $i$th agent's estimates of $\vartheta_{\max} \triangleq \max\{\vartheta_1, \ldots, \vartheta_n\}$.

*Lemma 4:* For any strongly connected digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the max-consensus algorithm in (19) guarantees that the agents reach consensus on $\vartheta_{\max}$ in finite time.

*Proof:* See [26, Proposition 17]. ∎

Though (18) only provides bound on individual entries of $\dot{\boldsymbol{\phi}}(t)$, a global bound on $\dot{\boldsymbol{\phi}}(t)$ can be obtained as

$$\sup_{t\in[t_0,\infty)} \left\|\dot{\boldsymbol{\phi}}(t)\right\|_\infty \leq \vartheta_{\max}. \tag{20}$$

*Remark 2:* Similar to the max-consensus algorithm in (19), a min-consensus algorithm can be easily implemented by replacing the $\mathrm{sgn}_+$ operator in (19) with $\mathrm{sgn}_-$, where $\mathrm{sgn}_- : \mathbb{R} \mapsto \mathbb{R}$ is defined as $\mathrm{sgn}_-(x) = -1$ if $x < 0$ and zero otherwise. The finite-time convergence property of the min-consensus algorithm follows.

Max-consensus can also be used to estimate a bound on the network size, i.e., the number of agents in the network [29], [30]. At first, each agent locally generates $m \geq 10$ uniformly distributed independent random variables $w_{i,k}$ between 0 and 1, $i \in \mathcal{I}$ and $k = 1, \ldots, m$. Agents then distributedly compute $w_{\max,k}$, the maximum of each $m$ random variable across the entire network using the following max-consensus algorithm:

$$\dot{\hat{w}}_{i,k}(t) = \mathrm{sgn}_+ \left(\sum_{j\in\mathcal{N}_i} a_{ij}\left(\hat{w}_{j,k}(t) - \hat{w}_{i,k}(t)\right)\right) \tag{21}$$

where $\hat{w}_{i,k}(t_0) = w_{i,k}$, $i \in \mathcal{I}$, and $k = 1, \ldots, m$. Now, each agent obtains an upper bound on $n$ as follows:

$$\hat{n}_i(t) = -2m\left(\sum_{k=1}^{m} \log\left(\hat{w}_{i,k}(t)\right)\right)^{-1}. \tag{22}$$

Now, based on the results given in [30], after the agents reach max-consensus on $w_{i,k}$, we have $\hat{n}_1(t) = \hat{n}_2(t) = \cdots = \hat{n}_n(t) = \hat{n}_{\max} \geq n$, for all $t \geq t_0 + t^*$, where $t^*$ is the maximum time it takes for agents to reach max-consensus on $m$ random variables. Now, it follows from (20) that $\forall\, i \in \mathcal{I}$, we have

$$\|\dot{\boldsymbol{\phi}}(t)\|_2 \leq \hat{\vartheta}_i(t)\sqrt{\hat{n}_i(t)} \quad \forall\, t \geq t_0 + t^*. \tag{23}$$

Here, we assume each agent has its own performance criteria $\epsilon_i > 0$, i.e., the DAC error for the $i$th agent is required to be less than $\frac{\epsilon_i}{\sigma_2(\mathcal{L})}$, where $\sigma_2(\mathcal{L})$ denotes the minimum nonzero singular value of $\mathcal{L}$. The DAC estimator is designed such that the most stringent among these performance criteria is satisfied. Therefore, the proposed DAC algorithm requires the agents to implement the following min-consensus algorithm $\forall i \in \mathcal{I}$:

$$\dot{\varepsilon}_i(t) = \mathrm{sgn}_-\left(\sum_{j\in\mathcal{N}_i} a_{ij}\left(\varepsilon_j(t) - \varepsilon_i(t)\right)\right), \quad \varepsilon_i(t_0) = \epsilon_i. \tag{24}$$

Based on Lemma 4, the agents reach min-consensus in finite time. Let $t^*$ be the time it takes for the agents to reach min-consensus. Thus, $\forall\, i \in \mathcal{I}$ and $\forall\, t \geq t_0 + t^*$, we have

$$\varepsilon_i(t) = \varepsilon^\star \triangleq \min\{\epsilon_1, \ldots, \epsilon_n\}.$$

The technical results given in this section are organized as follows. Since the agents only have access to the local difference in the average-consensus error and not the error signal itself, Lemma 5 first proves that bounding the average-consensus error difference among the agents within a strongly connected balanced digraph indeed bounds the consensus error signal itself if the error signal satisfies the zero-sum condition. In Theorem 3, inputs to the robust DAC estimator are selected such that the consensus error difference among the agents is globally ultimately bounded. The zero-sum condition on the average-consensus error is proven in Theorem 4. Finally, Theorem 5 combines the results from Lemma 5 and Theorems 3 and 4 to establish the global ultimate boundedness of the average-consensus error.

*Lemma 5:* For a strongly connected weight-balanced digraph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, if there exists a vector $\mathbf{x} \in \mathbb{R}^n$ such that $\sum_{i=1}^{n} x_i = 0$ and $\|\mathcal{L}\mathbf{x}\|_2 \leq \epsilon$ for some $\epsilon > 0$, then $\|\mathbf{x}\|_2 \leq \frac{\epsilon}{\sigma_2(\mathcal{L})}$, where $\sigma_2(\mathcal{L})$ denotes the minimum nonzero singular value of $\mathcal{L}$.

*Proof:* Note that $\|\mathcal{L}\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathcal{L}^\top \mathcal{L}\mathbf{x}}$. For a strongly connected digraph, $\mathcal{L}^\top\mathcal{L}$ is a positive-semidefinite matrix with a single eigenvalue at zero, corresponding to the eigenvector $\mathbf{1}_n$. Let $0 < \lambda_2(\mathcal{L}^\top\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L}^\top\mathcal{L})$ be the eigenvalues of $\mathcal{L}^\top\mathcal{L}$, and let $\mathbf{1}_n, \boldsymbol{\nu}_2(\mathcal{L}^\top\mathcal{L}), \ldots, \boldsymbol{\nu}_n(\mathcal{L}^\top\mathcal{L})$ be the corresponding eigenvectors. From the Courant–Fischer theorem [31], we have, $\forall\, i \in \mathcal{I}$,

$$\lambda_i\left(\mathcal{L}^\top\mathcal{L}\right) = \min_{\substack{\|\mathbf{x}\|_2 \neq 0 \\ \boldsymbol{\nu}_j^\top \mathbf{x} = 0,\ \forall j \in\{1,\ldots,i-1\}}} \frac{\mathbf{x}^\top\mathcal{L}^\top\mathcal{L}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}}.$$

Thus, we have $\lambda_2(\mathcal{L}^\top\mathcal{L}) = \min_{\substack{\|\mathbf{x}\|_2\neq 0\\ \mathbf{1}_n^\top\mathbf{x}=0}} \frac{\mathbf{x}^\top\mathcal{L}^\top\mathcal{L}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}}$. Since $\mathbf{1}_n^\top\mathbf{x} = 0$, we have $\mathbf{x} \notin \mathbb{R}^n_{\mathbf{1}} \setminus \{\mathbf{0}_n\}$. Therefore, $\lambda_2(\mathcal{L}^\top\mathcal{L})\|\mathbf{x}\|_2^2 \leq \mathbf{x}^\top\mathcal{L}^\top\mathcal{L}\mathbf{x}$ and $\sqrt{\mathbf{x}^\top\mathcal{L}^\top\mathcal{L}\mathbf{x}} \geq \sqrt{\lambda_2(\mathcal{L}^\top\mathcal{L})\|\mathbf{x}\|_2^2} = \sigma_2(\mathcal{L})\|\mathbf{x}\|_2$. Thus, we have $\sigma_2(\mathcal{L})$ $\|\mathbf{x}\|_2 \leq \epsilon$. ∎

In essence, Lemma 5 proves that if $\mathbf{1}^\top\mathbf{x} = 0$, then $\|\mathcal{L}\mathbf{x}\|_\infty \leq \epsilon$ implies $\|\mathbf{x}\|_2 \leq \frac{\epsilon}{\sigma_2(\mathcal{L})}$. We rely on this result to show global ultimate boundedness of the DAC error.

### B. Robust DAC Algorithm

Here, we propose the following robust DAC algorithm:

$$\dot{q}_i(t) = \eta_i(t), \quad q_i(t_0) \tag{25}$$

$$p_i(t) = \phi_i(t) + \sum_{j\in\mathcal{N}_i} a_{ij}\left(q_i(t) - q_j(t)\right) \tag{26}$$

where $q_i(t)$, $p_i(t)$, and $\eta_i(t)$ denote the internal state, each node's estimate of $\bar{\phi}(t)$, and the estimator input, respectively. The input $\eta_i(t)$ is to be designed. In vector form, we have

$$\dot{\mathbf{q}}(t) = \boldsymbol{\eta}(t), \quad \mathbf{q}(t_0) = \mathbf{q}_0 \tag{27a}$$

$$\mathbf{p}(t) = \mathcal{L}\,\mathbf{q}(t) + \boldsymbol{\phi}(t). \tag{27b}$$

From (27b), we have

$$\dot{\mathbf{p}}(t) = \mathcal{L}\,\boldsymbol{\eta}(t) + \dot{\boldsymbol{\phi}}(t). \tag{28}$$

Let $\tilde{\mathbf{p}}(t) \triangleq \mathbf{p}(t) - \bar{\phi}(t)\mathbf{1}_n$ denote the DAC error; now, the error dynamics can be written as

$$\dot{\tilde{\mathbf{p}}}(t) = \mathcal{L}\,\boldsymbol{\eta}(t) + M\dot{\boldsymbol{\phi}}(t). \tag{29}$$

Before presenting the main technical results, we define the consensus error difference among the nodes as

$$\boldsymbol{\mu}(t) \triangleq \mathcal{L}\,\tilde{\mathbf{p}}(t) = \mathcal{L}\,\mathbf{p}(t). \tag{30}$$

After substituting (29), $\dot{\boldsymbol{\mu}}(t)$ can be written as

$$\dot{\boldsymbol{\mu}}(t) = \mathcal{L}\,\mathcal{L}\,\boldsymbol{\eta}(t) + \mathcal{L}\,\dot{\boldsymbol{\phi}}(t). \tag{31}$$

Note that the nodes only have access to $\boldsymbol{\mu}(t)$ and not the consensus error signal itself. Thus, the objective is to design $\boldsymbol{\eta}(t)$, based on $\boldsymbol{\mu}(t)$, such that $\boldsymbol{\eta}(t)$ forces the agents to reach consensus on $\tilde{\mathbf{p}}(t)$ .

*Theorem 3:* Given Assumptions 4 and 5, the robust DAC estimator in (27) guarantees that the consensus error difference $\boldsymbol{\mu}(t)$ is globally ultimately bounded if, $\forall\, i \in \mathcal{I}$, the individual estimator inputs $\eta_i(t)$ are selected as

$$\eta_i(t) = -\frac{\hat{n}_i(t)\left(\hat{\vartheta}_i(t) + \delta\right)}{\varepsilon_i(t)} p_i(t) \tag{32}$$

where $\delta > 0$ is a design parameter; $\hat{\vartheta}_i(t)$, $\hat{n}_i(t)$, and $\varepsilon_i(t)$ are given in (19), (22), and (24), respectively. More specifically, we have $\|\boldsymbol{\mu}(t)\|_2 \le \varepsilon^\star$ for all $t \ge t_0 + t^\star + t^\diamond$, where $t^\star$ is the time it takes for the agents to reach max/min-consensus and $t^\diamond$ is given as

$$t^\diamond = \frac{\sqrt{2\boldsymbol{\mu}^\top(t_0+t^\star)\,(\mathcal{L})^+\,\boldsymbol{\mu}(t_0+t^\star)\sigma_{\max}\left((\mathcal{L})^+\right)}}{\delta\,\hat{n}_{\max}} \tag{33}$$

where $\sigma_{\max}(\cdot)$ denotes the maximum singular value.

*Proof:* Based on Assumption 5, $\dot{\boldsymbol{\phi}}(t)$ is bounded for all $t$. Furthermore, $\varepsilon_i(t)$, $\hat{\vartheta}_i(t)$, and $\hat{n}_i(t)$ are strictly positive and upper bounded. Now, boundedness of $\mathbf{p}(t)$ follows from the bounded-input bounded-output stability of (28). Thus, $\boldsymbol{\mu}(t)$ remains bounded for all $t \ge t_0$. The goal of this proof is not to merely show the boundedness of $\boldsymbol{\mu}(t)$, but to demonstrate that $\boldsymbol{\mu}(t)$ can be made arbitrarily small in finite time. Agents have the freedom to select this arbitrary bound, $\varepsilon^\star$, independent of any bounds given in Assumption 5 as long as $\varepsilon^\star > 0$.

For all $t_1 \ge t_0 + t^\star$, $\eta_i(t_1)$ for all $i \in \mathcal{I}$ can be written as

$$\eta_i(t_1) = -\frac{\hat{n}_{\max}\left(\vartheta_{\max} + \delta\right)}{\varepsilon^\star}\, p_i(t_1). \tag{34}$$

Define $V(t_1) = \frac{1}{2}\boldsymbol{\mu}^\top(t_1)(\mathcal{L})^+\boldsymbol{\mu}(t_1)$. Note that $V(t_1) > 0$ for all $\boldsymbol{\mu}(t_1) \ne \mathbf{0}$ and $V(t_1) = 0$ if and only if $\boldsymbol{\mu}(t_1) = \mathbf{0}$. Now, we have

$$\begin{aligned}\dot{V}(t_1) &= \boldsymbol{\mu}^\top(t_1)(\mathcal{L})^+\mathcal{L}\,\mathcal{L}\,\boldsymbol{\eta}(t_1) + \boldsymbol{\mu}^\top(t_1)(\mathcal{L})^+\mathcal{L}\,\dot{\boldsymbol{\phi}}(t_1),\\ &= \boldsymbol{\mu}^\top(t_1)\mathcal{L}\,\boldsymbol{\eta}(t_1) + \boldsymbol{\mu}^\top(t_1)\dot{\boldsymbol{\phi}}(t_1),\\ &\le \boldsymbol{\mu}^\top(t_1)\mathcal{L}\,\boldsymbol{\eta}(t_1) + \|\boldsymbol{\mu}(t_1)\|_2\|\dot{\boldsymbol{\phi}}(t_1)\|_2,\\ &\le \boldsymbol{\mu}^\top(t_1)\mathcal{L}\,\boldsymbol{\eta}(t_1) + \sqrt{n}\|\boldsymbol{\mu}(t_1)\|_2\|\dot{\boldsymbol{\phi}}(t_1)\|_\infty.\end{aligned}$$

Substituting (34) yields

$$\dot{V}(t_1) \le -\frac{\hat{n}_{\max}\left(\vartheta_{\max}+\delta\right)}{\varepsilon^\star}\|\boldsymbol{\mu}(t_1)\|_2^2 + \sqrt{n}\|\boldsymbol{\mu}(t_1)\|_2\|\dot{\boldsymbol{\phi}}(t_1)\|_\infty.$$

If $\|\boldsymbol{\mu}(t_1)\|_2 \ge \varepsilon^\star$, then $\frac{\|\boldsymbol{\mu}(t_1)\|_2^2}{\varepsilon^\star} \ge \|\boldsymbol{\mu}(t_1)\|_2$. Also, $\vartheta_{\max} \ge \|\dot{\boldsymbol{\phi}}(t_1)\|_\infty$ and $\hat{n}_{\max} \ge n$. Thus, we have $\dot{V}(t_1) \le -\delta\,\hat{n}_{\max}\|\boldsymbol{\mu}(t_1)\|_2$. Note that

$$\dot{V}(t_1) \le -\delta\,\hat{n}_{\max}\|\boldsymbol{\mu}(t_1)\|_2 = -\delta\,\hat{n}_{\max}\sqrt{\boldsymbol{\mu}^\top(t_1)\boldsymbol{\mu}(t_1)}$$

and

$$2V(t_1) = \boldsymbol{\mu}^\top(t_1)(\mathcal{L})^+\boldsymbol{\mu}(t_1) \le \sigma_{\max}\left(\mathcal{L}^+\right)\|\boldsymbol{\mu}(t_1)\|_2^2.$$

Thus, we have

$$\dot{V}(t_1) \le -\frac{\delta\,\hat{n}_{\max}\sqrt{2}}{\sqrt{\sigma_{\max}\left(\mathcal{L}^+\right)}}\sqrt{V(t_1)}.$$

Define $\varrho \triangleq \frac{\delta\,\hat{n}_{\max}\sqrt{2}}{\sqrt{\sigma_{\max}(\mathcal{L}^+)}}$. Now, we have $\frac{1}{2\sqrt{V(t_1)}}\dot{V}(t_1) \le -\frac{1}{2}\varrho$. Based on the comparison lemma (see [32, Lemma 3.4]), we have

$$\sqrt{V(t_1)} \le \sqrt{V(t_0+t^\star)} - \frac{1}{2}\varrho t_1. \tag{35}$$

Since $V(t_1)$ is a positive-definite function of $\boldsymbol{\mu}(t_1)$ for all $t_1 \ge t_0 + t^\star$ and $\dot{V}(t_1)$ is a negative-definite function of $\boldsymbol{\mu}(t_1)$ for all $\|\boldsymbol{\mu}(t_1)\|_2 \ge \varepsilon^\star$, we have $\|\boldsymbol{\mu}(t_1)\|_2 \le \varepsilon^\star$ for all $t_1 \ge t_0 + t^\star + t^\diamond$, where

$$t^\diamond = \frac{\sqrt{2V(t_0+t^\star)\sigma_{\max}\left(\mathcal{L}^+\right)}}{\delta\,\hat{n}_{\max}}. \tag{36}$$

■

Define $\rho(t) \triangleq \mathbf{1}_n^\top\tilde{\mathbf{p}}(t)$. The zero-sum condition on the average-consensus error, $\tilde{\mathbf{p}}(t)$, is proven next.

*Theorem 4:* Given Assumptions 4 and 5, the robust DAC algorithm in (27) guarantees that $\rho(t) = 0$ for all $t \ge t_0$.

*Proof:* From (27b), we have $\tilde{\mathbf{p}}(t) = \mathcal{L}\,\mathbf{q}(t) + M\boldsymbol{\phi}(t)$. Thus, $\rho(t) = \mathbf{1}_n^\top\tilde{\mathbf{p}}(t) = 0, \quad \forall\, t \ge t_0$. Also, from (29), we have $\dot{\rho}(t) = \mathbf{1}_n^\top\dot{\tilde{\mathbf{p}}}(t) = 0$. ■

The main result of the section is presented next.

*Theorem 5:* Given Assumptions 4 and 5, the robust DAC algorithm in (27) with estimator inputs given in (32) guarantees that the DAC error is globally ultimately bounded with ultimate bound $\frac{\varepsilon^\star}{\sigma_2(\mathcal{L})}$, i.e., $\|\tilde{\mathbf{p}}(t)\|_2 \le \frac{\varepsilon^\star}{\sigma_2(\mathcal{L})}$ for all $t \ge t_0 + t^\star + t^\diamond$, where $t^\diamond$ is given in (33).

*Proof:* From Theorem 3, we have $\|\boldsymbol{\mu}(t)\|_2 = \|\mathcal{L}\tilde{\mathbf{p}}(t)\|_2 \le \varepsilon^\star$ for all $t \ge t_0 + t^\star + t^\diamond$, and from Theorem 4, we have $\mathbf{1}_n^\top\tilde{\mathbf{p}}(t) = 0, \quad \forall\, t \ge t_0$. Thus, from the Courant–Fischer theorem, we have

$$\min_{\|\tilde{\mathbf{p}}(t)\|_2 \ne 0} \frac{\tilde{\mathbf{p}}^\top(t)\mathcal{L}^\top\mathcal{L}\tilde{\mathbf{p}}(t)}{\tilde{\mathbf{p}}^\top(t)\tilde{\mathbf{p}}(t)} = \lambda_2\left(\mathcal{L}^\top\mathcal{L}\right).$$

Therefore, $\|\mathcal{L}\tilde{\mathbf{p}}(t)\|_2^2 \ge \lambda_2\left(\mathcal{L}^\top\mathcal{L}\right)\|\tilde{\mathbf{p}}(t)\|_2^2$, and it follows from Lemma 5 that $\|\tilde{\mathbf{p}}(t)\|_2 \le \frac{\varepsilon^\star}{\sigma_2(\mathcal{L})}$ for all $t \ge t_0 + t^\star + t^\diamond$. ■

*Remark 3:* Note that the internal state, $\mathbf{q}(t)$, in (27) can grow unbounded for certain types of $\mathbf{p}(t)$. However, there exists two methods to address this internal stability issue [33], [34]. The first method given in [33] proposes a nonlinear mapping of the internal state to a compact manifold and replacing the linear Laplacian operator, $\mathcal{L}$, with a nonlinear Laplacian map that has a bounded image. The second approach given in [34] introduces an additional Laplacian in the internal dynamics, i.e., $\dot{\mathbf{q}} = \mathcal{L}\boldsymbol{\eta}$. George *et al.* [34] then propose a singular perturbation scheme to remove the communication burden induced by the consecutive Laplacian matrices in the algorithm.

*Remark 4:* Though the current formulation assumes that the network topology is static, the proposed algorithm in (27) also works for networks with slowly switching topology as long as the time in-between

TABLE I
EXISTING DAC ALGORITHMS FOR DIRECTED NETWORKS

| | Algorithm | Initialization Requirement | Steady-state bound on DAC error |
|---|---|---|---|
| Ref. [1] | $\dot{q}_i(t) = -\sum_{j\in\mathcal{N}_i} a_{ij}\left(p_i(t) - p_j(t)\right)$ <br> $p_i(t) = \phi_i(t) + q_i(t)$ | $\mathbf{q}(t_0) = \mathbf{0}$ | $\lim_{t\to\infty} \lvert\tilde{p}_i(t)\rvert \leq$ <br> $\sup_{t\in[t_0,\infty)} \lVert M\dot{\boldsymbol{\phi}}(t)\rVert_2 \left(\hat{\lambda}_2\right)^{-1}$ |
| Ref. [6] | $\dot{q}_i(t) = -\sum_{j\in\mathcal{N}_i} a_{ij}\left(f(p_i(t)) - f(p_j(t))\right)$ <br> $p_i(t) = \phi_i(t) + q_i(t)$ | $\mathbf{q}(t_0) = \mathbf{0}$ | $\lim_{t\to\infty} \lVert\tilde{\mathbf{p}}(t)\rVert_2 \leq$ <br> $\sup_{t\in[t_0,\infty)} \lVert\dot{\boldsymbol{\phi}}(t)\rVert_\infty \frac{\sqrt{2n}}{C\hat{\lambda}_2}$ |
| Ref. [8] | $\dot{q}_i(t) = \alpha\beta \sum_{j\in\mathcal{N}_i} a_{ij}\left(p_i(t) - p_j(t)\right)$ <br> $\dot{r}_i(t) = -\alpha r_i(t) - \beta \sum_{j\in\mathcal{N}_i} a_{ij}\left(p_i(t) - p_j(t)\right) - q_i(t)$ <br> $p_i(t) = \phi_i(t) + r_i(t)$ | $\mathbf{q}(t_0) = \mathbf{0}$ | $\lim_{t\to\infty} \lvert\tilde{p}_i(t)\rvert \leq$ <br> $\sup_{t\in[t_0,\infty)} \lVert M\dot{\boldsymbol{\phi}}(t)\rVert_2 \left(\frac{1}{\beta\,\hat{\lambda}_2}\right)$ |
| Proposed (27) | $\dot{q}_i(t) = \eta_i(t)$ <br> $p_i(t) = \phi_i(t) + \sum_{j\in\mathcal{N}_i} a_{ij}\left(q_i(t) - q_j(t)\right)$ | None | $\forall t \geq t_0 + t^\star + t^\diamond$ <br> $\lVert\tilde{\mathbf{p}}(t)\rVert_2 \leq \frac{\varepsilon^\star}{\sigma_2(\mathcal{L})}$ |

two consecutive switches is always greater than the convergence time of the algorithm, i.e., $t^\star + t^\diamond$.

### C. Performance Comparison

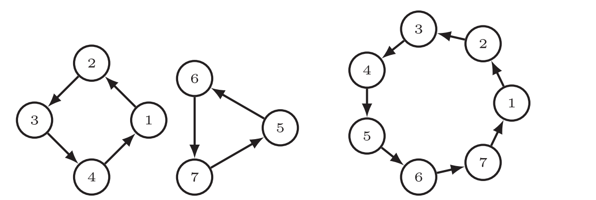
In this subsection, we compare the performance of the robust DAC algorithm in (27) to existing continuous-time algorithms for directed networks in [1], [6], and [8]. A summary of these existing algorithms, any initialization requirement, and their steady-state DAC error is given in Table I.[2] Compared to the existing work, the proposed algorithm has no specific initialization requirement and, therefore, is robust to initialization errors. Also, the proposed algorithm guarantees finite-time convergence to a prescribed bound, while the algorithms in [1], [6], and [8] only provide asymptotic boundedness, and the convergence bound depends on a global bound on the time derivative of the reference signal.

Compared to the robust DAC algorithms for undirected networks given in Section IV, algorithm in (27) cannot guarantee asymptotic convergence of the DAC error. Theoretically, the robust algorithm in Fig. 2 has proven to yield zero consensus error. However, precise implementation of the discontinuous switching signal is impossible in practice due to the finite machine precision, and therefore, zero steady-state error is unattainable in reality. The boundary-layer approach, which approximates the switching function by a smooth continuous function inside a boundary layer, is typically used to implement the switching function in practice. With the boundary-layer approximation, $\mathbf{y}(t) = B^\top\tilde{\mathbf{x}}(t)$ converges to an $\varepsilon$-neighborhood of the origin in finite time. Therefore, following the same arguments given in the proof of Theorem 5, we can easily see that the performance of the robust algorithm in Fig. 2 with the boundary-layer approximation is similar to that of the robust algorithm for the directed network proposed here.

## VI. NUMERICAL SIMULATION

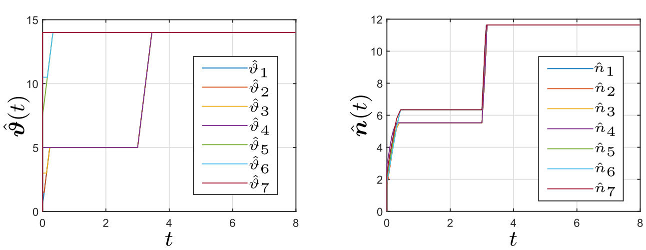
Consider a directed network of seven nodes given in Fig. 3. Initially, there are two disconnected subgraphs, as shown in Fig. 3(a), and at $t = 3$, the two disconnected components join together to form a single connected graph [see Fig. 3(b)].

[2]Here, $\hat{\lambda}_2 = \lambda_2(\mathcal{L} + \mathcal{L}^\top)$, and $f(\cdot) : \mathbb{R} \mapsto \mathbb{R}$ is a continuous and strictly increasing function.




(a) (b)

Fig. 3. Communication graph of seven networked agents. (a) During $0 \leq t < 3$. (b) During $3 \leq t$.




(a) (b)

Fig. 4. Finite-time convergence of the max-consensus algorithm in (19) and the network size estimation algorithms in (22). (a) Max-consensus on $\vartheta_i$. (b) Network size estimate.

Here, we select unit link weights and sinusoidal reference signals. The individual reference signals are given as

$$\phi_i(t) = a_i \sin(\omega_i t + \psi_i)) + 7 \quad \forall\, i \in \{1, \ldots, 4\}$$
$$\phi_i(t) = a_i \cos(\omega_i t + \psi_i)) - 5 \quad \forall\, i \in \{5, 6, 7\}$$

where $a_i = i$, $\omega_i = \frac{1}{4}(i + 1)$, and $\psi_i(t) = \frac{2\pi i}{7} - \pi$. For simulation purposes, we select $\delta = 1$ and $\epsilon_i = 10^{-5}$ for all $i$. We also select $m = 10$ and $\vartheta_i = a_i\omega_i$ for all $i$.

Fig. 4 contains $\hat{\vartheta}_i(t)$ and $\hat{n}_i(t)$ for individual nodes obtained from implementing the max-consensus algorithms in (19) and (22), respectively. Fig. 4 indicates that the agents are able to reach consensus on $\vartheta_{\max}$ and $\hat{n}_{\max}$ in finite time.

Fig. 5(a) gives the average of the reference signals $\bar{\phi}_1(t)$ and $\bar{\phi}_2(t)$, denoted using thick dotted line, and the individual node estimates of the averages. Note that initially there are two different averages, each associated with one of the two subgraphs, and for all $t \geq 3$, there is a

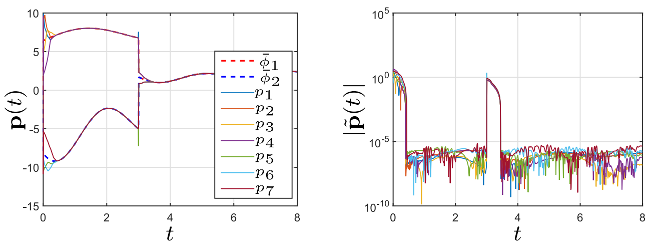



Fig. 5. Consensus estimates and errors for the robust DAC algorithm. (a) Robust DAC estimates. (b) Consensus error.

single average corresponding to the combined graph. Fig. 5(b) gives the magnitude of average-consensus error for individual nodes. Note that the consensus error in Fig. 5(b) reaches the prescribed bound in finite time.

## VII. Conclusion

Here, we considered the DAC problem and proposed three algorithms that are robust to initialization errors. Two of the proposed algorithms in Section IV are concerned with undirected networks and make use of an adaptive scheme that removes the explicit dependence of the algorithm parameters on any upper bounds on the reference signal or its time derivatives. The third algorithm presented in Section V provides a robust solution to the DAC problem on directed networks. Compared to the existing algorithms for directed networks, the proposed robust algorithm guarantees that the consensus error converges to an $\varepsilon^\star$-neighborhood of the origin in finite time. The current formulation also allows each node to select its own performance criteria for the DAC problem. Utilizing a max/min-consensus algorithm, the proposed approach then selects the appropriate algorithm parameters distributedly such that the most stringent performance requirement among the nodes is satisfied. This eliminates the need for any adaptive gains, as well as any global bounds on the reference signals. Performance comparison of the proposed algorithm to existing techniques and the numerical simulation of the robust algorithm are also presented. Future work includes developing event-triggered and privacy preserving versions of the proposed algorithm and applying them to distributed learning problems.

## References


[1] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005, Paper We-A18-TO/1.

[2] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *Proc. 45th IEEE Conf. Decis. Control*, Dec. 2006, pp. 338–343.

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

[4] H. Bai, "A two-time-scale dynamic average consensus estimator," in *Proc. 55th Conf. Decis. Control*, Dec. 2016, pp. 75–80.

[5] S. S. Kia, J. Cortés, and S. Martínez, "Singularly perturbed algorithms for dynamic average consensus," in *Proc. Eur. Control Conf.*, Jul. 2013, pp. 1758–1763.

[6] S. Nosrati, M. Shafiee, and M. B. Menhaj, "Dynamic average consensus via nonlinear protocols," *Automatica*, vol. 48, no. 9, pp. 2262–2270, 2012.

[7] S. S. Kia, J. Cortés, and S. Martínez, "Dynamic average consensus with distributed event-triggered communication," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 890–895.

[8] S. S. Kia, J. Cortés, and S. Martínez, "Dynamic average consensus under limited control authority and privacy requirements," *Int. J. Robust Nonlinear Control*, vol. 25, no. 13, pp. 1941–1966, 2015.

[9] S. S. Kia, J. Cortés, and S. Martínez, "Distributed event-triggered communication for dynamic average consensus in networked systems," *Automatica*, vol. 59, pp. 112–119, 2015.

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

[11] F. Chen, Y. Cao, and W. Ren, "Distributed computation of the average of multiple time-varying reference signals," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 1650–1655.

[12] 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.

[13] F. Chen and W. Ren, "Robust distributed average tracking for coupled general linear systems," in *Proc. 32nd Chin. Control Conf.*, Jul. 2013, pp. 6953–6958.

[14] F. Chen, G. Feng, L. Liu, and W. Ren, "An extended proportional-integral control algorithm for distributed average tracking and its applications in Euler-Lagrange systems," in *Proc. Amer. Control Conf.*, Jun. 2014, pp. 2581–2586.

[15] S. Ghapani, W. Ren, and F. Chen, "Distributed average tracking for double-integrator agents without using velocity measurements," in *Proc. Amer. Control Conf.*, Jul. 2015, pp. 1445–1450.

[16] F. Chen, G. Feng, L. Liu, and W. Ren, "Distributed average tracking of networked Euler-Lagrange systems," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 547–552, Feb. 2015.

[17] F. Chen, W. Ren, W. Lan, and G. Chen, "Distributed average tracking for reference signals with bounded accelerations," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 863–869, Mar. 2015.

[18] S. Ghapani, S. Rahili, and W. Ren, "Distributed average tracking for second-order agents with nonlinear dynamics," in *Proc. Amer. Control Conf.*, Jul. 2016, pp. 4636–4641.

[19] Y. Zhao, Y. Liu, Z. Li, and Z. Duan, "Distributed average tracking for multiple signals generated by linear dynamical systems: An edge-based framework," *Automatica*, vol. 75, pp. 158–166, 2017.

[20] S. Rahili and W. Ren, "Heterogeneous distributed average tracking using nonsmooth algorithms," in *Proc. Amer. Control Conf.*, May. 2017, pp. 691–696.

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

[22] W. X. I. Gutman, "Generalized inverse of the Laplacian matrix and some applications," *Bull., Classe des Sci. Math. Naturelles, Sci. Math.*, vol. 129, no. 29, pp. 15–23, 2004.

[23] P. Ioannou and J. Sun, *Robust Adaptive Control*. New York, NY, USA: Dover, 2013.

[24] A. Filippov, *Differential Equations With Discontinuous Righthand Sides: Control Systems* (ser. Mathematics and its Applications). Dordrecht, The Netherlands: Springer, 1988.

[25] E. Lavretsky, T. E. Gibson, and A. M. Annaswamy, "Projection operator in adaptive systems," 2012, arXiv ePrints, arXiv:1112.4232v6.

[26] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[27] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 6103–6107, Nov. 2012.

[28] S. Giannini, D. D. Paola, A. Petitti, and A. Rizzo, "On the convergence of the max-consensus protocol with asynchronous updates," in *Proc. 52nd IEEE Conf. Decis. Control*, Dec. 2013, pp. 2605–2610.

[29] H. Terelius, D. Varagnolo, and K. H. Johansson, "Distributed size estimation of dynamic anonymous networks," in *Proc. 51st IEEE Conf. Decis. Control*, Dec. 2012, pp. 5221–5227.

[30] D. Varagnolo, G. Pillonetto, and L. Schenato, "Distributed cardinality estimation in anonymous networks," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 645–659, Mar. 2014.

[31] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[32] H. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, chs. 3/11.

[33] B. V. Scoy, R. A. Freeman, and K. M. Lynch, "A fast robust nonlinear dynamic average consensus estimator in discrete time," in *Proc. 5th IFAC Workshop Distrib. Estimation Control Netw. Syst.*, Sep. 2015, pp. 191–196.

[34] J. George, R. A. Freeman, and K. M. Lynch, "Robust dynamic average consensus algorithm for signals with bounded derivatives," in *Proc. Amer. Control Conf.*, May 2017, pp. 352–357.