Adaptive Filter Algorithms for Accelerated Discrete-Time Consensus

Renato L. G. Cavalcante, *Member, IEEE*, and Bernard Mulgrew, *Senior Member, IEEE*

***Abstract*—In many distributed systems, the objective is to reach agreement on values acquired by the nodes in a network. A common approach to solve such problems is the iterative, weighted linear combination of those values to which each node has access. Methods to compute appropriate weights have been extensively studied, but the resulting iterative algorithms still require many iterations to provide a fairly good estimate of the consensus value. In this study we show that a good estimate of the consensus value can be obtained with few iterations of conventional consensus algorithms by filtering the output of each node with set-theoretic adaptive filters. We use the adaptive projected subgradient method to derive a set-theoretic filter requiring only local information available to each node and being robust to topology changes and erroneous information about the network. Numerical simulations show the good performance of the proposed method.**

***Index Terms*—Adaptive filters, consensus algorithms, distributed estimation.**

I. Introduction

In networks where each node has knowledge of a quantity of interest, reaching agreement in every node is often necessary. (See [1] for an overview of many applications.) We say that agreement (or consensus) is achieved when all nodes arrive at a weighted sum of the initial values acquired by the nodes in the network (in the following this weighted sum is referred to as *consensus value*). Usually, there is no single node with access to all information in the network, and each node can communicate only with its neighbors. Therefore, a great deal of effort has been devoted to the study of consensus algorithms in distributed networks [1]–[16].

In discrete-time systems, a conventional method to reach consensus is the iterative, weighted linear combination of all values to which a node has access: its own value and those from its neighbors. By representing the network as a graph, the weights are the values assigned to the edges of the graph, and they can be expressed as the entries of what is commonly referred to as the network matrix. If this matrix satisfies certain properties [3], the resulting iterative algorithm guarantees that each node obtains a sequence of estimates converging to the consensus value asymptotically. Such matrices are usually obtained with methods based on the Laplacian matrix of the network graph [1], but the convergence speed of the resulting iterative algorithm is often poor. To improve the converge rate, we can cast the choice of the network matrix as the solution of a convex optimization problem [3]. However, solving such optimization problems in distributed networks is not a trivial task or may require global knowledge of the network topology in every node.

If the above-mentioned linear iterative algorithms are used, each node can compute the consensus value in finite time by storing and processing only its own sequence of estimates [7], [9]. In [7], the consensus value is obtained in each node with a linear combination of the local past estimates. If every node has information about the network topology, the optimal linear combination of past values is readily available. However, knowledge of the topology in every node is a strong assumption, so a decentralized method to compute the optimal linear combination has also been devised in [7]. This decentralized approach requires many re-initializations of the original consensus algorithm with a special set of initial conditions. Alternatively, we can also run in parallel many instances of the original consensus algorithm with different initial conditions, but in this case the amount of data transmitted at each iteration increases. An additional limitation of this approach is that it has not been proven to be robust against topology changes during the iteration process. In turn, the method in [9] requires neither the transmission of additional information nor re-initializations of the algorithm. It uses the scalar epsilon algorithm (SEA) applied to the sequence of estimates in each node. The main disadvantages of this approach are that it uses all previous estimates and its robustness against changes in the network topology is questionable [15]. In addition, the estimated consensus value may fail to exist or may become numerically large in some iterations [17].

Motivated by the work in [7], we propose a method in which each node estimates the consensus value by filtering the output of conventional iterative consensus algorithms with set-theoretic adaptive filters. No reinitialization with different initial conditions is necessary, and combining the method with other efficient schemes, such as that in [3], is also possible. Consensus is not guaranteed to be achieved in finite time, but we show that every node is able to improve greatly the estimate of the consensus value with very few iterations in many practical situations. In addition, we also prove that the method is robust against link failures and model mismatches, although in such cases there may be no gain in applying the set-theoretic filters. Therefore, the proposed method should be used in situations where the network is supposed to be fixed, but the algorithm should not collapse in the event of a link failure. We derive an efficient *decentralized* set-theoretic filter based on the

Manuscript received December 08, 2008; accepted August 10, 2009. First published September 18, 2009; current version published February 10, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Roberto Lopez-Valcarce.

The authors are with the Digital Communications Research Institute, Joint Research Institute for Signal and Image Processing, The University of Edinburgh, Edinburgh EH9 3JL, U.K. (e-mail: rlgc@ecs.soton.ac.uk; b.mulgrew@ed.ac.uk).

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/TSP.2009.2032450

adaptive projected subgradient method [18], [19]. Unlike many applications involving adaptive filters and diffusion networks, where the objective is usually to use consensus algorithms to improve the performance of adaptive filters [20], [21], here the objective is to use adaptive filters to improve the performance of consensus algorithms. To the best of the authors' knowledge, the latter is also a new application of adaptive filter techniques.

Fixed, nonadaptive filtering techniques have also been shown to improve the convergence speed of conventional consensus algorithms [6], [11], [15], [16]. However, unlike the method proposed here, the main idea of these techniques is to reshape the eigenvalues of the network matrix in such a way that the worst case convergence speed is improved. These algorithms modify the estimates of the consensus value exchanged by the nodes, and some of them are not robust against link failures.

The paper is divided as follows. In Section II we review basic concepts of convex analysis, the adaptive projected subgradient method, and the basic idea of conventional consensus algorithms. In Section III-A we rederive some of the results in [7] using basic linear algebra theory, and we also prove that set-theoretic filters can be used to estimate the consensus value with low estimation error. In Section III-B we give an example of a set-theoretic filter derived from the adaptive projected subgradient method. We study in Section III-C the robustness of the algorithm against networks with time-varying topologies and model mismatches. Numerical examples in Section IV show the performance of the algorithm.

II. Preliminaries

A. Basic Concepts in Convex Analysis

For every vector $\boldsymbol{v} \in \mathbb{R}^N$, we define the norm of $\boldsymbol{v}$ by $\|\boldsymbol{v}\| := \sqrt{\boldsymbol{v}^T\boldsymbol{v}}$, which is the norm induced by the Euclidean inner product $\langle \boldsymbol{v}, \boldsymbol{y} \rangle := \boldsymbol{v}^T\boldsymbol{y}$ for every $\boldsymbol{v}, \boldsymbol{y} \in \mathbb{R}^N$. For a matrix $\boldsymbol{X} \in \mathbb{R}^{M\times N}$, its spectral norm is $\|\boldsymbol{X}\|_2 := \max\{\sqrt{\lambda}|\lambda$ is an eigenvalue of $\boldsymbol{X}^T\boldsymbol{X}\}$, which satisfies $\|\boldsymbol{X}\boldsymbol{y}\| \leq \|\boldsymbol{X}\|_2\|\boldsymbol{y}\|$ for any vector $\boldsymbol{y}$ of compatible size. A set C is said to be convex if $\boldsymbol{v} = \nu\boldsymbol{v}_1 + (1-\nu)\boldsymbol{v}_2 \in C$ for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $0 \leq \nu \leq 1$ [22], [23]. Let $C \subset \mathbb{R}^N$ be a nonempty closed convex set. The metric projection $P_C : \mathbb{R}^N \to C$ maps $\boldsymbol{v} \in \mathbb{R}^N$ to the uniquely existing vector $P_C(\boldsymbol{v}) \in C$ satisfying $\|\boldsymbol{v} - P_C(\boldsymbol{v})\| = \min_{\boldsymbol{y}\in C}\|\boldsymbol{v}-\boldsymbol{y}\| =: d(\boldsymbol{v}, C)$.

A function $\Theta : \mathbb{R}^N \to \mathbb{R}$ is said to be *convex* if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\forall \nu \in [0,1]$, $\Theta(\nu\boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leq \nu\Theta(\boldsymbol{x}) + (1-\nu)\Theta(\boldsymbol{y})$ (in this case Θ is continuous at every point in $\mathbb{R}^N$). The *subdifferential* of Θ at $\boldsymbol{y}$ is the nonempty closed convex set of all the *subgradients* of Θ at $\boldsymbol{y}$:

$$\partial\Theta(\boldsymbol{y}) := \{\boldsymbol{a} \in \mathbb{R}^N | \Theta(\boldsymbol{y}) + \langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{a}\rangle \leq \Theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^N\}.$$

B. Adaptive Projected Subgradient Method

Fact 1: Adaptive Projected Subgradient Method (APSM) [19].

Let $\Theta[i] : \mathbb{R}^N \to [0,\infty) \quad (\forall i \in \mathbb{N})$ be a sequence of convex functions and $\mathcal{C} \subset \mathbb{R}^N$ a nonempty closed convex set. For an arbitrarily given $\boldsymbol{h}[0] \in \mathcal{C}$, the adaptive projected subgradient method produces a sequence $(\boldsymbol{h}[i])_{i\in\mathbb{N}}$ by

$$\boldsymbol{h}[i+1] := \begin{cases} P_{\mathcal{C}}\left(\boldsymbol{h}[i] - \mu[i]\frac{\Theta[i](\boldsymbol{h}[i])}{\|\Theta'[i](\boldsymbol{h}[i])\|^2}\Theta'[i](\boldsymbol{h}[i])\right) \\ \qquad \text{if} \quad \Theta'[i](\boldsymbol{h}[i]) \neq \boldsymbol{0} \\ \boldsymbol{h}[i] \quad \text{otherwise} \end{cases} \tag{1}$$

where $\Theta'[i](\boldsymbol{h}[i]) \in \partial\Theta[i](\boldsymbol{h}[i])$, $\partial\Theta[i](\boldsymbol{h}[i])$ is the subdifferential of $\Theta[i]$ at $\boldsymbol{h}[i]$, and $0 \leq \mu[i] \leq 2$. We quote from [19] some selected properties of the algorithm.

a) *Monotone approximation*—Suppose

$$\boldsymbol{h}[i] \notin \Omega[i] := \left\{\boldsymbol{h} \in \mathcal{C} | \Theta[i](\boldsymbol{h}) = \Theta^\star[i] := \inf_{\boldsymbol{h}\in\mathcal{C}}\Theta[i](\boldsymbol{h})\right\} \neq \emptyset.$$

Then, for every $\mu[i] \in (0, 2(1 - (\Theta^\star[i])/(\Theta[i](\boldsymbol{h}[i]))))$, we have $(\forall \boldsymbol{h}^\star[i] \in \Omega[i])\|\boldsymbol{h}[i+1] - \boldsymbol{h}^\star[i]\| < \|\boldsymbol{h}[i] - \boldsymbol{h}^\star[i]\|$.

b) *Boundedness, Asymptotic optimality*—Suppose

$$\exists N_0 \in \mathbb{N} \quad \text{s.t.} \begin{cases} \Theta^\star[i] = 0, \forall i \geq N_0 \quad \text{and} \\ \Omega := \cap_{i\geq N_o}\Omega[i] \neq \emptyset. \end{cases}$$

Then $(\boldsymbol{h}[i])_{i\in\mathbb{N}}$ is bounded. Moreover, if we specially use $\forall \mu[i] \in [\epsilon_1, 2-\epsilon_2] \subset (0,2)$, we have $\lim_{i\to\infty}\Theta[i](\boldsymbol{h}[i]) = 0$ provided that $(\Theta'[i](\boldsymbol{h}[i]))_{i\in\mathbb{N}}$ is bounded. The parameters $\epsilon_1, \epsilon_2 > 0$ are arbitrarily small positive (real) numbers used to guarantee that no subsequence of $\mu[i]$ converges to zero or two.

For other properties, including the convergence of $(\boldsymbol{h}[i])_{i\in\mathbb{N}}$, the reader is referred to [19].

C. Distributed Consensus

Consider a network with undirected graph given by $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} := \{1, \ldots, N\}$ is the node set and $\mathcal{E}$ is the edge set [1], [3], an unordered pair of nodes. If nodes $k \in \mathcal{N}$ and $l \in \mathcal{N}$ can exchange information, then $\{k,l\} \in \mathcal{E}$ (unlike some references, here we assume that $\{k,k\} \in \mathcal{E}$ for every $k \in \mathcal{N}$). The set of neighbors of node k is defined by $\mathcal{N}_k := \{l \in \mathcal{N} | \{l,k\} \in \mathcal{E}\}$.

Let $x_k[0] \in \mathbb{R}$ be an initial value known by node k. An important class of iterative diffusion protocols produces a sequence $x_k[i]$ in every node k converging to asymptotic consensus:

$$\lim_{i\to\infty} x_k[i] = \frac{1}{N}\sum_{l\in\mathcal{N}} x_l[0], \quad \forall k \in \mathcal{N}. \tag{2}$$

Such a sequence is obtained with the following linear combination:

$$x_k[i+1] = \sum_{l\in\mathcal{N}_k} \omega_{kl} x_l[i], \quad \forall k \in \mathcal{N} \tag{3}$$

where $\omega_{kl} = \omega_{lk}$ is a weight satisfying certain properties described later in this section and associated with the edge $\{l,k\}$. If $\{l,k\} \notin \mathcal{E}$, in the following we assume that $\omega_{kl} = 0$. We can equivalently rewrite (3) as

$$\boldsymbol{x}[i+1] = \boldsymbol{W}\boldsymbol{x}[i] \tag{4}$$

where $\boldsymbol{x}[i] = [x_1[i] \dots x_N[i]]^T \in \mathbb{R}^N$ and the component of the lth row and kth column of $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is ω_{lk}. In connected and undirected graphs,[1] there always exists a choice of $\boldsymbol{W}$ such that (2) is satisfied for every $\boldsymbol{x}[0] \in \mathbb{R}^N$ with the iteration in (4), [1], [3], [8]. In such graphs, $\boldsymbol{W}$ satisfies the following properties $||\boldsymbol{W} - 1/N\mathbf{1}_N\ \mathbf{1}_N^T||_2 < 1$, $\boldsymbol{W}^T = \boldsymbol{W}$, and $\boldsymbol{W}\mathbf{1}_N = \mathbf{1}_N$ [3], [8], where $\mathbf{1}_N \in \mathbb{R}^N$ is the vector of ones. The value $||\boldsymbol{W} - 1/N\ \mathbf{1}_N\mathbf{1}_N^T||_2$ is a measure of the worst-case convergence speed, so a great deal of effort has been devoted to devising efficient methods to minimize $||\boldsymbol{W} - 1/N\ \mathbf{1}_N\mathbf{1}_N^T||_2$ [3]. Usually, these algorithms can only guarantee asymptotic convergence, and the convergence speed can still be poor when compared to many other existing techniques [5]–[7], [15]. In particular, the work in [7] proved that achieving consensus in a finite number of iterations with time-invariant topologies is possible. If the nodes do not have knowledge about the network matrix $\boldsymbol{W}$, in each node k the algorithm in [7] computes the consensus value from samples $x_k[i]$ obtained with (4) using different initial vectors $\boldsymbol{x}[0]$. Motivated by this work, in the next section we show filters that can greatly improve the estimate of the consensus value in each node without using different initial vectors $\boldsymbol{x}[0]$.

III. Proposed Algorithm

Before deriving efficient set-theoretic filters, we first prove through the use of a line matrix, defined in Section III-A, that there exists a finite impulse response filter that can compute the consensus value in every node from the values obtained with the iteration in (3) (cf. Theorem 1).[2] Then, also in Section III-A, we show that, if a node does not have enough information to compute an optimal filter, a sequence of filters satisfying certain properties can be used to estimate the consensus value with possibly low estimation error (cf. Theorem 2). Section III-B shows that such a sequence can be generated with the adaptive projected subgradient method, and Section III-C proves that the filters are robust against topology changes and/or erroneous information about the network.

A. On the Existence of Linear Filters

Definition 1: A matrix $\boldsymbol{W}$ is said to be compatible with the graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$ if ω_{lk}, the component of the kth column and lth row of $\boldsymbol{W}$, satisfies $\omega_{lk} = 0$ whenever $\{l, k\} \notin \mathcal{E}$.

Definition 2: The line matrix $\boldsymbol{L}_\beta \in \mathbb{R}^{N\times N} (\beta \in \mathbb{R})$ of $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ is the matrix defined by $\boldsymbol{L}_\beta := \boldsymbol{I} + \beta(\boldsymbol{W} - \boldsymbol{I})$.[3]

The name line matrix stems from the fact that, if we treat the matrices $\boldsymbol{W}$ and $\boldsymbol{I}$ as points in an N^2 dimensional Hilbert space, then a line matrix of $\boldsymbol{W}$ is a point lying on the line passing through both $\boldsymbol{W}$ and $\boldsymbol{I}$. An important property of line matrices is that, if $\boldsymbol{W}$ is a matrix compatible with the graph $\mathcal{G}$, the line matrix $\boldsymbol{L}_\beta$ of $\boldsymbol{W}$ is also compatible with the graph $\mathcal{G}$ for every $\beta \in \mathbb{R}$.

Lemma 1: Let $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ be a matrix compatible with the graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $1 < N = |\mathcal{N}|$ ($|\mathcal{N}|$ denotes the cardinality of the set $\mathcal{N}$). Assume that $\boldsymbol{W}$ satisfies the conditions for average consensus in undirected graphs: $||\boldsymbol{W} - 1/N\mathbf{1}_N\mathbf{1}_N^T||_2 < 1$, $\boldsymbol{W}^T = \boldsymbol{W}$, and $\boldsymbol{W}\mathbf{1}_N = \mathbf{1}_N$. Then there exists a sequence of line matrices $\boldsymbol{L}_{\beta_0}, \dots, \boldsymbol{L}_{\beta_{T-1}}$ of $\boldsymbol{W}$ such that the iteration $\boldsymbol{x}[i+1] = \boldsymbol{L}_{\beta_i}\boldsymbol{x}[i]$ reaches consensus in finite time: $\boldsymbol{x}[T] = (1/N)\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{x}[0]$ for every $\boldsymbol{x}[0] \in \mathbb{R}^N$, where T is the number of distinct eigenvalues of $\boldsymbol{W}$, excluding the eigenvalue one.

Proof: The proof is given in Appendix I. ∎

Lemma 1 can be easily extended to directed graphs as long as $\boldsymbol{W}$ is nondefective (in such a case, we can also consider complex matrices and/or weighted-average consensus). From an implementation point of view, using a line matrix $\boldsymbol{L}_\beta$ of $\boldsymbol{W}$ instead of $\boldsymbol{W}$ itself amounts to changing the weights that node k assigns to its neighbors $x_l[i] (l \in \mathcal{N}_k)$ in the iteration in (3). Therefore, if the topology is known by all nodes, they can use the above lemma to compute the consensus value in finite time by exchanging and keeping only current estimates $x_k[i]$. In contrast, the result in [7, Sect. III]—which uses the final value theorem of the z-transform—requires from each node also the storage of past values $x_k[i]$. Knowledge of the topology in every node is a strong assumption, so in the following we focus on the case where the nodes do not have more information other than an upper bound of the number of nodes, and the resulting algorithm also requires storage of past values $x_k[i]$.

The following theorem shows that, to compute the consensus value in finite time, we can alternatively filter $x_k[i]$ obtained with (4) in each node instead of using a sequence of line matrices.

Theorem 1 (On the Existence of Linear Filters): Assume the conditions in Lemma 1. In addition, let $M \in \mathbb{N}$ be a fixed integer with $M \geq T + 1$. For every $k \in \mathcal{N}$, define $\boldsymbol{y}_k[i] \in \mathbb{R}^M$ by $\boldsymbol{y}_k[i] := [x_k[i] \dots x_k[i - M + 1]]^T \in \mathbb{R}^M$ $(i \geq M - 1)$ (recall that $\boldsymbol{x}[i] = [x_1[i] \dots x_N[i]]^T$ and that $\boldsymbol{x}[i+1] = \boldsymbol{W}\boldsymbol{x}[i]$). Then the convex set $\mathcal{K}$ defined by

$$\mathcal{K} := \{\boldsymbol{h} \in \mathbb{R}^M | \boldsymbol{h}^T\boldsymbol{y}_k[i] = (1/N)\mathbf{1}_N^T\boldsymbol{x}[0], \quad \forall \boldsymbol{x}[0] \in \mathbb{R}^N, k \in \mathcal{N}\} \quad (5)$$

is nonempty. Intuitively, the set $\mathcal{K}$ is the set of linear filters $\boldsymbol{h} \in \mathbb{R}^M$ that can compute the consensus value $(1/N)\mathbf{1}_N^T\boldsymbol{x}[0]$ for every initial condition $\boldsymbol{x}[0] \in \mathbb{R}^N$ in every node by filtering any M consecutive samples of the local information $x_k[i]$.

Proof: The proof is given in Appendix II. ∎

(The previous theorem can be straighforwardly extended to the case where the matrix $\boldsymbol{W}$ is complex and nondefective.)

Theorem 2: Consider the assumptions in Theorem 1 and choose a matrix $\boldsymbol{W}$ satisfying the conditions for average consensus. Denote the consensus value by $v^\star := \lim_{i\to\infty} x_k[i] = (1/N)\mathbf{1}_N^T\boldsymbol{x}[0]$, where $\boldsymbol{x}[i] = [x_1[i] \cdots x_N[i]]^T \in \mathbb{R}^N$ is obtained with the iteration in (4). In node $k \in \mathcal{N}$, let $\boldsymbol{h}_k[i+1] \in \mathbb{R}^M$ be a sequence of filters used to estimate the consensus value $v^\star$ from $\boldsymbol{y}_k[i]$ for $i \geq M - 1$ according to $v_k[i] := \boldsymbol{h}_k[i+1]^T\boldsymbol{y}_k[i]$. The estimation error in node k at time $i \geq M - 1$ is defined by $|v^\star - v_k[i]|$. Assume that the following

[1]Connected graphs are graphs in which there is a path passing through all nodes.

[2]This proof has been first shown in [7]. Here we present an alternative derivation of some results in [7]. This alternative derivation is fundamental to the new results that follow.

[3]Line matrices with the restriction $0 \leq \beta \leq 1$ have been used in [11] (but the terminology "line matrix" has not been used). One of the main contributions of the work in [11] is the optimal choice (in some sense) of a *fixed* line matrix that should replace $\boldsymbol{W}$ in the iteration in (4).

conditions are satisfied: i) $\boldsymbol{h}_k[i]^T\mathbf{1}_M = 1$ at any time instant and ii) $\|\boldsymbol{h}_k[i+1] - \boldsymbol{h}^\star\| \leq \|\boldsymbol{h}_k[i] - \boldsymbol{h}^\star\|$ for every $\boldsymbol{h}^\star \in \mathcal{K}$. Then:

a) *Asymptotic optimality*—Consensus is achieved asymptotically, i.e., $\lim_{i\to\infty} v_k[i] = v^\star (k \in \mathcal{N})$.

b) *Upper bound of the estimation error*–For given $\boldsymbol{x}[0] \in \mathbb{R}^N$, an upper bound of the estimation error $|v_k[i] - v^\star|$ is $|v_k[i] - v^\star| \leq Sd(\boldsymbol{h}[i+1], \mathcal{K})$, where $S \in \mathbb{R}$ is an upper bound of the sequence $\|\boldsymbol{y}_k[i]\|$ associated with the initial condition $\boldsymbol{x}[0]$. (Note that $S < \infty$ because, due to the choice of $\boldsymbol{W}$, $\boldsymbol{y}_k[i]$ converges to $v^\star\mathbf{1}_M$). Therefore, as $\|\boldsymbol{h}[i] - \boldsymbol{h}^\star\|$ is monotone nonincreasing for every $\boldsymbol{h}^\star$ (condition ii)), the upper bound of the estimation error is also monotone nonincreasing.

Proof: The proof is given in Appendix III. ∎

Remark 1: On Theorem 2.

1) At time i, the estimate of the consensus value obtained with the original consensus algorithm in (3) is the same as that obtained by applying the filter $[1\ 0 \ldots 0]^T =: \boldsymbol{e}$ to $\boldsymbol{y}_k[i]$ in every node k. The upper bound of the estimation error given in Theorem 2(b) is monotone nonincreasing, so, if the first filter of the sequence is set to $\boldsymbol{e}$, then we see that the upper bound of the estimation error is always less or equal than that of the original consensus algorithm in (3). In addition, Theorem 2(a) shows that such an approach is asymptotically optimal, i.e., the estimation error converges to zero.
2) Note that there is no guarantee that estimate of the consensus value is improved with the proposed approach. In addition, the upper bound in Theorem 2(b) may not be tight depending on the initial filter $\boldsymbol{h}_k[i]$. However, in the following we show that, by devising a sequence of filters with decreasing distance to every $\boldsymbol{h}^\star \in \mathcal{K}$ and with $\boldsymbol{h}_k[i]$ satisfying properties also met by any $\boldsymbol{h}^\star \in \mathcal{K}$, the estimate of the consensus value is improved in many practical cases. Intuitively, we can expect good estimates because, in each iteration, $\boldsymbol{h}_k[i]$ tries to reproduce more accurately the behavior of $\boldsymbol{h}^\star \in \mathcal{K}$, the filter able to compute the consensus value in finite time. Set-theoretic filtering is a common approach to devise such filters $\boldsymbol{h}_k[i]$.

B. Set-Theoretic Adaptive Filters for Distributed Consensus

As discussed in Remark 1, by filtering the output of conventional consensus algorithms in (3), the estimate of the consensus value is asymptotically optimal and potentially improved in every iteration. Here we show how to produce a sequence $\boldsymbol{h}_k[i]$ satisfying conditions i) and ii) of Theorem 2 using only local information available to each node. The basic idea is to design supersets of $\mathcal{K}$ (constructed with local information and corresponding to the expected behavior of any $\boldsymbol{h}^\star \in \mathcal{K}$) and use the adaptive projected subgradient method to generate a sequence $\boldsymbol{h}_k[i]$ approaching the intersection of the supersets.

Condition i) of Theorem 2 defines the following hyperplane:

$$C := \{\boldsymbol{h} \in \mathbb{R}^M | \boldsymbol{h}^T\mathbf{1}_M = 1\}, \tag{6}$$

a closed convex set. We will prove later that every $\boldsymbol{h}^\star \in \mathcal{K}$ also belongs to C (cf. Lemma 2). To find a closed convex set defined by the local information $\boldsymbol{y}_k[i]$ (see Theorem 1), we first note that $(\boldsymbol{h}^\star)^T\boldsymbol{y}_k[i] = v^\star$, which corresponds to a hyperplane requiring knowledge of the consensus value $v^\star = (1/N)\sum_{k\in\mathcal{N}} x_k[0]$. To remove the dependence on $v^\star$, we use the difference vector

$$\tilde{\boldsymbol{\delta}}_k[i,n] := \boldsymbol{y}_k[i] - \boldsymbol{y}_k[n]$$

which satisfies $(\boldsymbol{h}^\star)^T\tilde{\boldsymbol{\delta}}_k[i,n] = 0$ for every $i, n \geq M-1$, $k \in \mathcal{N}$, and $\boldsymbol{h}^\star \in \mathcal{K}$. In the following, we normalize $\tilde{\boldsymbol{\delta}}_k[i,n]$ to mitigate numerical problems when $\boldsymbol{y}_k[i] - \boldsymbol{y}_k[n]$ approaches zero. More precisely, we use

$$\boldsymbol{\delta}_k[i,n] := \frac{(\boldsymbol{y}_k[i] - \boldsymbol{y}_k[n])M}{\|\boldsymbol{y}_k[i] - \boldsymbol{y}_k[n]\| + \epsilon} \tag{7}$$

where $\epsilon > 0$ is arbitrarily small. Note that the equality $(\boldsymbol{h}^\star)^T\boldsymbol{\delta}_k[i,n] = 0$ still holds with this normalization. Therefore, for $i, n \geq M-1$, we can also use hyperplanes defined by

$$\mathcal{D}_k[i,n] := \{\boldsymbol{h} \in \mathbb{R}^M | \boldsymbol{h}^T\boldsymbol{\delta}_k[i,n] = 0\} \supset \mathcal{K} \tag{8}$$

which are fully characterized with only local available information.

Lemma 2: For the iteration $\boldsymbol{x}[i+1] = \boldsymbol{W}\boldsymbol{x}[i]$ with $\boldsymbol{W}$ satisfying the conditions for average consensus, we have that $\mathcal{K} \subset \mathcal{D}_k[n,i] \cap C \neq \emptyset$ for every $i, n \geq M-1$ and $k \in \mathcal{N}$, where the scalar M is as defined in Theorem 1.

Proof: Theorem 1 shows that $\mathcal{K}$ in (5) is nonempty. Thus, by definition, any $\boldsymbol{h}^\star \in \mathcal{K}$ satisfies $(\boldsymbol{h}^\star)^T\boldsymbol{y}_k[i] = (1/N)\sum_{k\in\mathcal{N}} x_k[0]$ for any initial condition $\boldsymbol{x}[0] = [x_1[0] \ldots x_N[0]]^T$ and $k \in \mathcal{N}$. In particular, let $\boldsymbol{x}[0] = \mathbf{1}_N$. By construction, $\boldsymbol{W}$ satisfies $\boldsymbol{W}\mathbf{1}_N = \mathbf{1}_N$, which shows that $\boldsymbol{y}_k[i] = \mathbf{1}_M$ for $i \geq M-1$ and $k \in \mathcal{N}$. Therefore, $(\boldsymbol{h}^\star)^T\boldsymbol{y}_k[i] = (\boldsymbol{h}^\star)^T\mathbf{1}_M = 1$, and thus $\mathcal{K} \subset C$, which, together with (8), shows that $C \cap \mathcal{D}_k[i,n] \supset \mathcal{K} \neq \emptyset$. ∎

The proposed algorithm approaches (asymptotically) a point in the intersection of C and a suitable selection of sets $\mathcal{D}_k[i,n]$. The next theorem shows that, condition ii) of Theorem 2 can be satisfied with set-theoretic filters derived from the adaptive projected subgradient method applied to a sequence of cost functions based on C and $\mathcal{D}_k[i,n]$.

Theorem 3: In node k and at time i, define a convex function $\Theta_k[i] : \mathbb{R}^M \to [0,\infty)$ such that $\Theta_k(\boldsymbol{h}) = 0$ whenever $\boldsymbol{h}$ belongs to C and a suitable selection of sets $\mathcal{D}_k[i,n] (i, n \geq M)$. Applying the adaptive projected subgradient method to $\Theta_k[i]$ with a step size within the range $\mu_k[i] \in (0,2)$, we have that $\|\boldsymbol{h}_k[i+1] - \boldsymbol{h}^\star\| \leq \|\boldsymbol{h}_k[i] - \boldsymbol{h}^\star\|$ for every $\boldsymbol{h}^\star \in \mathcal{K}$. In addition, if the step size satisfies $\mu_k[i] \in [\epsilon_1, 2-\epsilon_2] \subset (0,2)$ and the subgradients $\Theta_k'[i](\boldsymbol{h}_k[i])$ produced by the iteration in (1) are bounded, then $\lim_{i\to\infty} \Theta_k[i](\boldsymbol{h}_k[i]) = 0$.

Proof: From Lemma 2, we know that every filter $\boldsymbol{h}^\star \in \mathcal{K}$ belongs to the intersection of any combination of sets $\mathcal{D}_k[i,n]$ and C. Consequently, we have that $\Theta_k[i](\boldsymbol{h}^\star) = 0$, and the proof is an immediate consequence of Fact 1(a)–(b) in Section II-B. ∎

Note that, once cost functions meeting the properties in Theorem 3 are designed, we can set $\mathcal{C}$ in (1) to $\mathcal{C} := C$ so that both conditions i) and ii) of Theorem 2 are satisfied with filters derived with the adaptive projected subgradient method. Next, we

give a particular sequence of cost functions, but many others are possible.

1) Constrained Affine Projection Algorithm (CAPA): At time $i \geq M$, define the following set, which combines sets $\mathcal{D}[n,k]$ and C at the same time:

$$V_k[i] := \{\boldsymbol{h} \in \mathbb{R}^M | \boldsymbol{H}_k[i]^T \boldsymbol{h} = [0 \dots 01]^T\} = \bigcap_{j=1}^{Q[i]} \mathcal{D}[i, i-j] \cap C \quad (9)$$

where $\boldsymbol{H}_k[i] := [\boldsymbol{\delta}_k[i,i-1]\boldsymbol{\delta}_k[i,i-2]\dots\boldsymbol{\delta}_k[i,i-Q[i]]\mathbf{1}_M] \in \mathbb{R}^{M\times(Q[i]+1)} (i \geq M), Q[i] = \min(i-M, J)$, and $J+1$ is the number of past vectors $\boldsymbol{y}_k[i]$ that should be stored in memory for the computation of $\boldsymbol{H}_k[i]$. (Lemma 2 shows that the set $V_k[i]$ is nonempty.)

Hence, at time $i \geq M$ and in node k, we can use the cost function $\Theta_k[i] : \mathbb{R}^M \to [0,\infty)$ defined by

$$\Theta_k[i](\boldsymbol{h}) = d(\boldsymbol{h}, V_k[i]). \quad (10)$$

A subgradient of the function in (10) is [19]

$$\partial\Theta_k[i](\boldsymbol{h}) \ni \Theta'_k[i](\boldsymbol{h}) = \begin{cases} \frac{\boldsymbol{h} - P_{V_k[i]}(\boldsymbol{h})}{d(\boldsymbol{h}, V_k[i])} & \text{if } \boldsymbol{h} \notin V_k[i] \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (11)$$

Applying (10) and (11) to the scheme in (1) with $\mathcal{C} = C$ (to force condition (i) of Theorem 2), after some simple manipulations we arrive at

$$\boldsymbol{h}_k[i+1] = P_C(\boldsymbol{h}_k[i] + \mu_k[i](P_{V_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i])) \quad (12)$$

where $k \in \mathcal{N}$, $\mu_k[i] \in (0,2)$ is the step size, $P_C(\boldsymbol{h}) = \boldsymbol{h} - (\boldsymbol{h}^T\mathbf{1}_M - 1)\mathbf{1}_M/\|\mathbf{1}_M\|^2$ is the projection of $\boldsymbol{h}$ onto C,

$$P_{V_k[i]}(\boldsymbol{h}) = \boldsymbol{h} - (\boldsymbol{H}_k[i]^\dagger)^T(\boldsymbol{H}_k[i]^T\boldsymbol{h} - [0 \ \dots \ 1]^T) \quad (13)$$

is the projection of $\boldsymbol{h}$ onto $V_k[i]$ [22], and $(\cdot)^\dagger$ denotes the Moore–Penrose pseudo-inverse [24, Sec. 5.5.4]. Note that the algorithm in (12) resembles the affine projection algorithm (APA) [19], [25]–[29].

In the computation of pseudo-inverses, we should treat as zero singular values less than some pre-defined threshold because the presence of very small singular values can cause numerical problems in finite precision arithmetic [24, p. 258].[4] A problem caused by this truncation is that the filters can be moved away from the set C when computing the projection onto $V_k[i] \supset C$. As shown in Theorem 2, under very mild conditions, filters belonging to C are asymptotically optimal, so in practice we should always use the projection P_C in (12), even if it is not theoretically necessary (e.g., when $\mu_k[i] = 1$).

The computational complexity of the algorithm is dominated by the computation of the SVD of $\boldsymbol{H}_k[i]$. Without exploiting any structure present in $\boldsymbol{H}_k[i]$, we can compute the SVD with $4M^2(Q[i]+1) + 8M(Q[i]+1)^2 + 9(Q[i]+1)^3$ flops[5] [24, Sec. 5.4.5], so in large networks the computational cost of the proposed method may be prohibitive. However, Section III-C shows that, if the algorithm can be applied, the consensus value is often obtained with great accuracy. In terms of memory requirements, we need to keep $M + Q[i]$ samples $x_k[i]$ in the memory of each node.

C. Robustness Against Topology Changes and/or Model Mismatches

So far we have assumed networks with fixed topology and knowledge of an upper bound for N in every node. However, in many scenarios (e.g., wireless sensor networks) links can fail due to reachability problems, jammers, moving obstacles, etc. [14]. The next theorem shows that, if one or both assumptions fail, the consensus value estimated by the proposed method is at least asymptotically optimal. This property contrasts with many existing algorithms, which do not guarantee the convergence of the estimate in time-varying networks (examples are given in [15]).

Theorem 4: Suppose that the iteration

$$\begin{bmatrix} x_1[i+1] \\ \vdots \\ x_N[i+1] \end{bmatrix} = \boldsymbol{x}[i+1] = \boldsymbol{W}[i]\boldsymbol{x}[i]$$

is able to reach consensus asymptotically, i.e., $\lim_{i\to\infty}\boldsymbol{x}[i] = (1/N)\mathbf{1}_N\ \mathbf{1}_N^T\boldsymbol{x}[0]$, where $\boldsymbol{W}[i] \in \mathbb{R}^{N\times N}$ is a time-varying matrix compatible with the network at time i.[6] Assume that the sequence of filters $\boldsymbol{h}_k[i] \in \mathbb{R}^S$ ($k \in \mathcal{N}$ and $S \in \mathbb{N}$) is bounded and satisfy $\boldsymbol{h}_k[i]^T\mathbf{1}_S = 1$. Then consensus is achieved asymptotically, i.e., $\lim_{i\to\infty} v_k[i] = (1/N)\sum_{k\in\mathcal{N}} x_k[0]$, where $v_k[i] := \boldsymbol{h}_k[i+1]^T\boldsymbol{y}_k[i]$ and $\boldsymbol{y}_k := [x_k[i]\ x_k[i-1]\dots x_k[i-S+1]]^T \in \mathbb{R}^S$.

Proof: The proof is almost identical to that of Theorem 2(a) and is omitted for brevity. Note that the length S of the vectors do not have any relation with the number of network nodes N. ■

Remark 2: On Theorem 4.

A subtle but important difference between Theorems 2 and 4 is the boundedness of the sequence $\boldsymbol{h}_k[i]$. In the former theorem, the boundedness is a consequence of the fact that the sequence $\{\|\boldsymbol{h}_k[i] - \boldsymbol{h}^\star\|\}$ (for fixed $\boldsymbol{h}^\star \in \mathcal{K}$) is monotone nonincreasing; in the latter, the boundedness is explicitly assumed because there may be no fixed filter $\boldsymbol{h}^\star$ that can provide the consensus value in every node for every initial condition. To force the boundedness of the sequence $\boldsymbol{h}_k[i]$ in every situation, we can replace (12) with

$$\boldsymbol{h}_k[i+1] = P_{C\cap\mathcal{B}}\left(\boldsymbol{h}_k[i] + \mu_k[i](P_{V_k[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i])\right) \quad (14)$$

where $\mathcal{B} := \{\boldsymbol{h} \in \mathbb{R}^M | \ \ \|\boldsymbol{h}\| \leq B\}$ is a closed convex set and $B > 0$ is an arbitrarily large number. However, as B is

[4] By noticing that $\lim_{i\to\infty}\boldsymbol{H}_k[i] = [\mathbf{0}_{M\times J}\mathbf{1}_M]$, where $\mathbf{0}_{M\times J} \in \mathbb{R}^{M\times J}$ is the matrix of zeros, we conclude that we cannot avoid the presence of arbitrarily small singular values.

[5] The projection onto $V_k[i]$ can be equivalently expressed and implemented in many different ways due to the presence of the pseudo-inverse in (12). Some of these alternative ways may avoid the computation of the SVD, but the numerical robustness of the algorithm against finite precision arithmetic and rank deficient matrices is usually decreased (see [30] and the references therein).

[6] This is not an unrealistic assumption [14].

arbitrarily large, in Section IV we show that in practice the set $\mathcal{B}$ in (14) may not be necessary.

IV. Numerical Simulations

A. Simulation Parameters

In the following, in each realization every node starts with a random number $x_k[0]$ with uniform distribution between 0 and 100.[7] The performance of interest is the global mean square error (MSE), defined by $\mathrm{MSE}[i] = \sum_{k\in\mathcal{N}} E[|v_k[i] - v^\star|^2]$, where $v^\star = (1/N)\sum_{k\in\mathcal{N}} x_k[0]$, $v_k[i] := \boldsymbol{h}_k[i+1]^T \boldsymbol{y}_k[i]$ for $i \geq M$, and $v_k[i] = x_k[i]$ for $i < M$ (because the filters are only defined for $i \geq M$). Ensemble average curves are obtained by averaging the results of 1000 realizations.

We compare the proposed CAPA algorithm with the eigenvalue shaping filter (ESF) in [6, eq. (27)] and indicate the parameters of the algorithms as follows.

- The CAPA algorithm is succeeded by two numbers separated by hyphens. The first number shows the filter length, and the second number is the value J in (9). The pseudo-inverse in (12) was computed with the pinv function of MATLAB (R2007a) with the default tolerance. In addition, the algorithm is initialized with the vector $[10\cdots0]^T \in \mathbb{R}^M$ in every node in every realization; the step size is $\mu_k[i]$ is set to one in every node; and the difference vectors in (7), which compose all but one column of $\boldsymbol{H}_k[i]$, use $\epsilon = 10^{-5}$.
- The ESF filter is succeeded by a number representing the value c in [6, eq. (27)].

Both the CAPA and ESF algorithms use same matrix $\boldsymbol{W}$ defined by

$$\boldsymbol{W} := \boldsymbol{I} + \rho\boldsymbol{\Phi} \tag{15}$$

where $\rho = 1/(\max_{k\in\mathcal{N}}(\phi_{kk}) + 10^{-5})$,

$$\phi_{kl} = \begin{cases} 1, & \text{if} \quad l \in \mathcal{N}_k \backslash \{k\} \\ -|\mathcal{N}_k| + 1, & \text{if} \quad l = k \\ 0, & \text{otherwise} \end{cases}$$

(ϕ_{kl} is the component of the kth row and lth column of the matrix $\boldsymbol{\Phi}$.) This matrix has been originally proposed in [6].

We also show the results obtained with the original consensus algorithm in (4) without any filters. For the iteration in (4), two matrices are considered: i) the matrix $\boldsymbol{W}$ used by the CAPA and ESF algorithms and ii) the optimal matrix $\boldsymbol{W}_{\mathrm{opt}}$, the solution of the convex optimization problem [3]

$$\begin{aligned} &\text{minimize } \left\|\boldsymbol{W} - \mathbf{1}_N\mathbf{1}_N^T/N\right\|_2 \\ &\text{subject to} \\ &\quad \boldsymbol{W} \in \mathcal{S} = \{\boldsymbol{W} \in \mathbb{R}^{N\times N} | w_{kl} = 0 \text{ if } (l,k) \notin \mathcal{E}\} \\ &\quad \boldsymbol{W}\mathbf{1}_N = \mathbf{1}_N \\ &\quad \boldsymbol{W} = \boldsymbol{W}^T, \end{aligned}$$

which was solved with `CVX`, a package for specifying and solving convex programs [31], [32].

Next, we show the performance of the algorithms with different parameters in different scenarios.

B. Static Networks

1) Rings: In Fig. 1, we show a network with a ring topology and the corresponding performance of the algorithms. The original iterative consensus algorithm in (4) with the matrix $\boldsymbol{W}$ defined in (15) and a ring topology is slow in general because the second largest eigenvalue (in magnitude) of $\boldsymbol{W}$ is close to one [4].[8] In such a case, we can see that the ESF-0.1 algorithm does not improve the estimate of the consensus value (by changing the parameter c, only marginal performance gains are obtained when compared to $c = 0.1$ in [6, eq. (27)]). In contrast, the set-theoretic filters greatly improve the estimate of the consensus value, providing even better performance than the original iterative algorithm with the optimal matrix $\boldsymbol{W}_{\mathrm{opt}}$. This is an important observation because the set-theoretic filters do not use global information about the network topology. Note that in this example the minimum filter length to satisfy the conditions of Theorem 1 (and thus of Theorem 3) is $M = 11$. Therefore, the filters of the CAPA algorithm in Fig. 1 have sufficient length.

In Fig. 2, we show the performance of the CAPA algorithm with different memory sizes and filter lengths. In Fig. 2(a) we fix the memory size and modify the filter length; in Fig. 2(b) we fix the filter length and modify the memory size. The parameters of the simulation are identical to those in Fig. 1.

For $M < 11$, the conditions for the existence of an optimal filter in Theorem 1 are violated, thus there may be no fixed filter able to provide the consensus value in every node for every initial value $x_k[0]$, $k \in \mathcal{N}$. Such a case is exemplified by the CAPA-4-3 algorithm in Fig. 2(a). However, as shown in Theorem 4, asymptotic convergence is expected (see also Remark 2), and this fact is also confirmed experimentally in Fig. 2(a). The CAPA-12-3 and CAPA-20-3 algorithms show that overestimating the filter length can reduce the convergence speed and increase the computational complexity.

As shown in Fig. 2(b), the performance of the algorithm can improve when more information is used at each iteration (i.e., when the memory size increases). By comparing the CAPA-20-19 and CAPA-20-9 algorithms, we see that in this example the former is using redundant sets $\mathcal{D}[i,n]$ as the performance of both algorithms is almost identical. However, the optimal memory size can be difficult to determine *a priori*, so as a general rule we should use as much information as possible. In practice, the computational power of each node limits the memory size.

2) Random Networks: We use a random network similar to that in [15], which in turn is based on the random geographic graph model in [33]. In each realization of the simulation, we randomly distribute 20 nodes with uniform distribution on a square grid with unit area. Nodes with distance less than $\sqrt{\log(N)/N}$ are neighboring nodes. We discard graphs not

[7] In our simulations, the relative performance of the compared algorithms was similar with different initialization assumptions (Gaussian random variables, etc.). Note that in this section a random variable and its realization have the same notation.

[8] Note that the second largest eigenvalue of $\boldsymbol{W}$ is a measure of the worst case convergence rate of the original consensus algorithm in (4), [3].

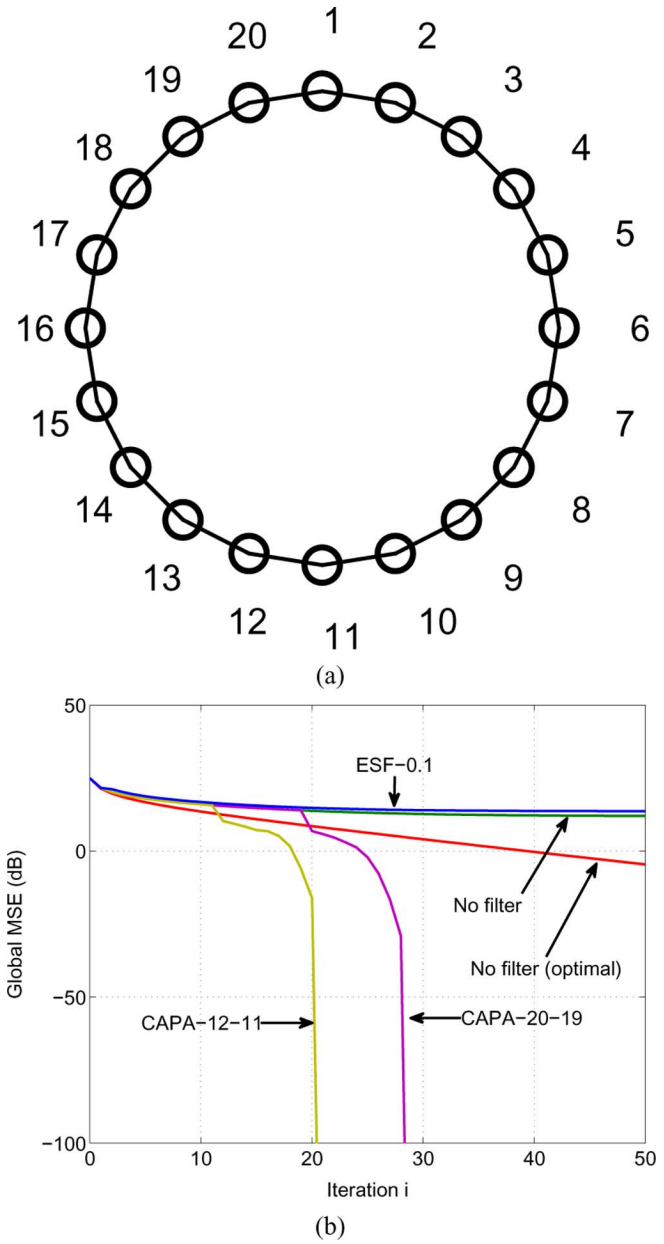

Fig. 1. Ring with 20 nodes. (a) Topology. (b) Transient performance of the algorithms.

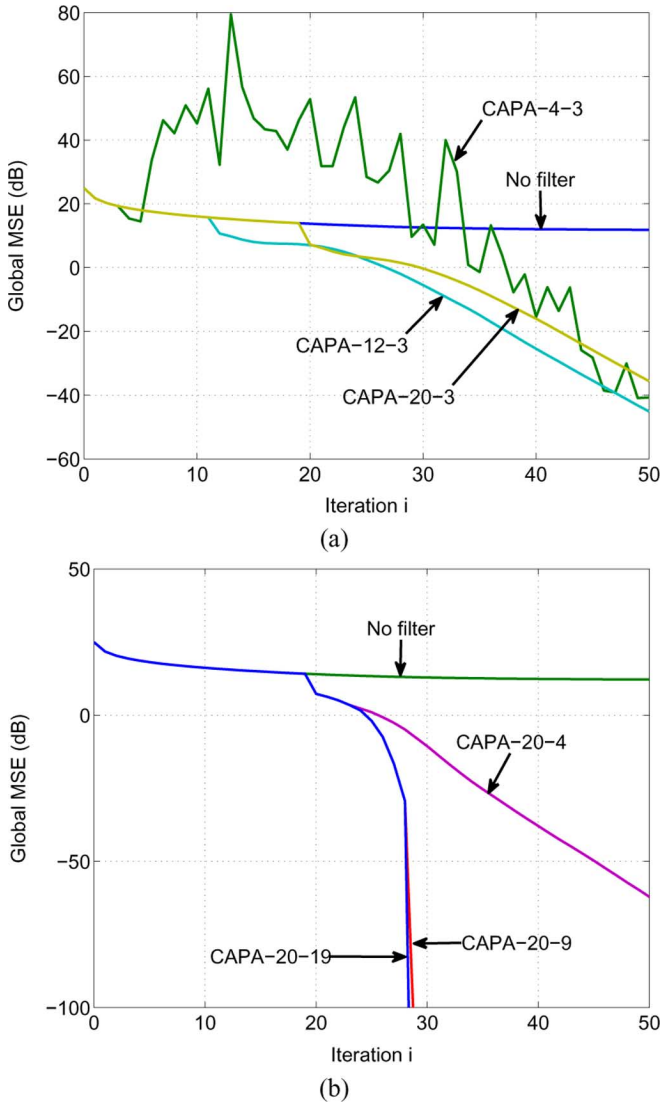

Fig. 2. Performance of the proposed algorithm in a network with 20 nodes and ring topology. (a) Fixed memory size and different filter lengths. (b) Fixed filter length and different memory sizes.

fully connected because consensus cannot be achieved in such cases. Fig. 3 shows the performance of the algorithms.

In this scenario, the ESF algorithm is able to improve greatly the estimate of the consensus value with low computational complexity. By increasing the parameter c in [6, eq. (27)], from 0.1 to 0.9 in 0.1 increments, the value $c = 0.8$ appeared to give the best performance for the ESF algorithm. However, the performance of the ESF algorithm is relatively sensitive to fairly small changes in c, and this fact can be seen by comparing the ESF-0.8 algorithm with the ESF-0.6 and ESF-0.7 algorithms. The ESF algorithm is based on a fixed filter, and a good value for c can be difficult to find without additional information about the network. When the network topology is unknown, the proposed algorithm is a good alternative because its adaptive nature can usually find good filters for any given network. Note

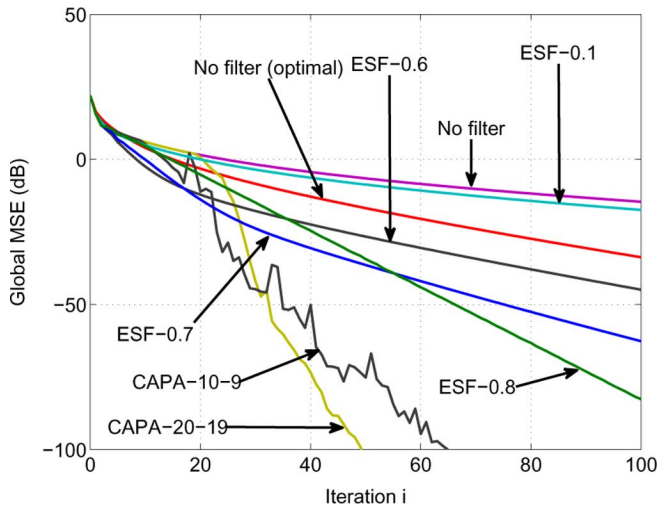

Fig. 3. Transient performance of the algorithms in a random network with 20 nodes.

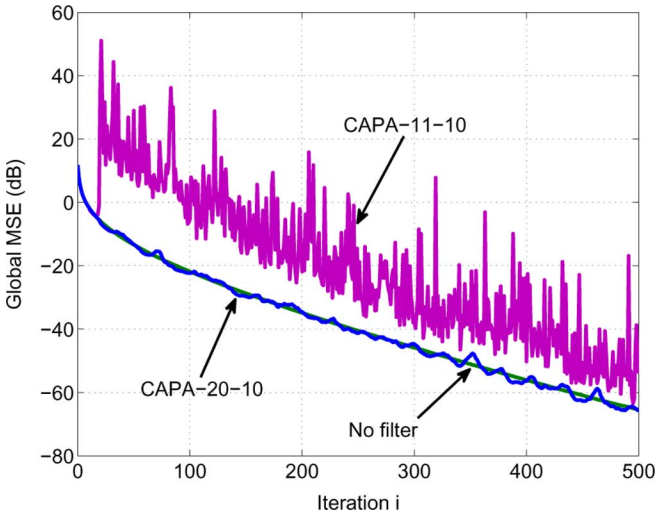

Fig. 4. Robustness against model mismatches.

that even the CAPA-10-9 algorithm, which uses a filter with not necessarily adequate length, can provide good performance.

C. Dynamic Networks

In Fig. 4, we show a random network with $N = 10$ nodes where in the first iteration the network topology is generated according to the description given in Section IV-B-2 (the topology changes in different realizations). In successive iterations, we assume that each link fails with probability 0.2. Link failures are equivalent to removing the corresponding edges of the graph, and those failures are independent in different iterations. In each iteration, the matrix $\boldsymbol{W}$ in (4) is replaced by a time-varying matrix $\boldsymbol{W}[i]$, recomputed according to (15) with the parameter ρ set to $\rho = 1/N$.[9] Fig. 4 shows the performance obtained by filtering the estimates $x_k[i]$ with the CAPA algorithm in each node. There may be no gain in applying the proposed algorithm in fast time-varying networks, but the estimates in all nodes approach the consensus value (see also Theorem 4 and Remark 2). Note that, in this scenario, some different methods for accelerated consensus do not guarantee convergence [9] (some methods even consider time-varying topologies). The robustness of the CAPA algorithm stems from the fact that the estimates exchanged by the nodes, which are expected to converge asymptotically, are not modified.

V. Final Remarks

We have shown that, by filtering the output of the original consensus algorithm in (4) with set-theoretic adaptive filters, an accurate estimate of the consensus value can be obtained with very few iterations. Therefore, schemes optimizing the network matrix $\boldsymbol{W}$, which often requires global information, may not be necessary. (However, the application of such schemes does not exclude their combination with set-theoretic filters). The adaptive filters are fully decentralized (the only information required by each node is an upper bound of the number of nodes in the network) and can provide good performance even with topologies and network matrices that do not work well with other existing methods. Finally, also unlike some existing approaches, the proposed algorithm is robust against topology changes and model mismatches.

[9]See [14] for the possible range of ρ. In the simulations in this section, the relative performance of the algorithms was not sensitive to the choice of ρ.

Appendix I
Proof of Lemma 1

Proof: Without loss of generality, assume that the eigenvalue decomposition of $\boldsymbol{W}$ is given by $\boldsymbol{W} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T$, where $\boldsymbol{D} = \mathrm{diag}(1, \lambda_1, \ldots, \lambda_{N-1})$ with eigenvalues λ_n in nonincreasing order. The matrix $\boldsymbol{Q}$ diagonalizes $\boldsymbol{W}$, hence it also diagonalizes the line matrix $\boldsymbol{L}_{\beta_i}$ for every $\beta_i \in \mathbb{R}$,:

$$\boldsymbol{L}_{\beta_i} = \boldsymbol{Q}(\boldsymbol{I} + \beta_i \mathrm{diag}(1, \lambda_1, \ldots, \lambda_{N-1}) - \beta_i \boldsymbol{I})\boldsymbol{Q}^T = \boldsymbol{Q}\boldsymbol{D}_{\beta_i}\boldsymbol{Q}^T$$

where $\boldsymbol{D}_{\beta_i} := \mathrm{diag}\,(1, 1+\beta_i(\lambda_1 - 1), \ldots, 1+\beta_i(\lambda_{N-1}-1)) \in \mathbb{R}^{N\times N}$. Define the set of distinct eigenvalues of $\boldsymbol{W}$ (in no particular order and excluding the eigenvalue one) by $\{\lambda_{l_0}, \ldots, \lambda_{l_{T-1}}\}$, and let $\beta_i = 1/(1-\lambda_{l_i})(i = 0, \ldots, T-1)$. Note that, for each time index i, we are setting to zero at least one different diagonal component of $\boldsymbol{D}_{\beta_i}$ (except for the first component, which is one by construction). Therefore, we have the following equality:

$$\boldsymbol{D}_{\beta_{T-1}} \cdots \boldsymbol{D}_{\beta_0} = \mathrm{diag}\,(1, 0, \ldots, 0).$$

By using $\boldsymbol{x}[i+1] = \boldsymbol{L}_{\beta_i}\boldsymbol{x}[i] = \boldsymbol{Q}\boldsymbol{D}_{\beta_i}\boldsymbol{Q}^T\boldsymbol{x}[i](i = 0, \ldots, T-1)$, we arrive at

$$\begin{aligned}\boldsymbol{Q}^T\boldsymbol{x}[T] &= \boldsymbol{D}_{\beta_{T-1}}\boldsymbol{Q}^T\boldsymbol{x}[T-1] \\ &= \boldsymbol{D}_{\beta_{T-1}} \cdots \boldsymbol{D}_{\beta_0}\boldsymbol{Q}^T\boldsymbol{x}[0] \\ &= \mathrm{diag}\,(1, 0, \ldots, 0)\boldsymbol{Q}^T\boldsymbol{x}[0]\end{aligned}$$

where in the second line we expanded the iteration $\boldsymbol{Q}^T\boldsymbol{x}[i+1] = \boldsymbol{D}_{\beta_i}\boldsymbol{Q}^T\boldsymbol{x}[i]$ recursively. From the above equation we conclude that $\boldsymbol{x}[T] = (1/N)\boldsymbol{1}_N\boldsymbol{1}_N^T\boldsymbol{x}[0]$ because, for $\boldsymbol{W}$, $\boldsymbol{1}_N/\sqrt{N}$ is the eigenvector associated with the eigenvalue $\lambda_0 = 1$. ■

Appendix II
Proof of Theorem 1

Proof: To prove that $\mathcal{K}$ is nonempty, we could use the results in [7]. However, in this study we use the concept of line matrices, and the proof that $\mathcal{K} \neq \emptyset$ comes naturally from basic linear algebra theory. Let the values $\beta_0, \ldots, \beta_{T-1}$ be a sequence such that consensus is achieved with the iteration $\boldsymbol{x}'[n+1] = \boldsymbol{L}_{\beta_i}\boldsymbol{x}'[n]$ in finite time for every $\boldsymbol{x}'[0] \in \mathbb{R}^N$, where $\boldsymbol{L}_{\beta_n}$ is a

line matrix of $\boldsymbol{W}$ (Lemma 1 guarantees the existence of such a sequence β_n). Without loss of generality, assume that $\boldsymbol{x}'[0] := \boldsymbol{W}^P\boldsymbol{x}[0] \in \mathbb{R}^N$, where $P \in \mathbb{N}$. Therefore,

$$\begin{aligned}
&(1/N)\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{x}'[0] \\
&\quad = \boldsymbol{x}'[T] \\
&\quad = \boldsymbol{L}_{\beta_{T-1}}\cdots\boldsymbol{L}_{\beta_0}\boldsymbol{x}'[0] \\
&\quad = (1/N)\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{W}^P\boldsymbol{x}[0] \\
&\quad = \boldsymbol{Q}\mathrm{diag}(1,0,\ldots,0)\boldsymbol{Q}^T\boldsymbol{W}^P\boldsymbol{x}[0] \\
&\quad = \boldsymbol{Q}\mathrm{diag}(1,0,\ldots,0)\boldsymbol{Q}^T\boldsymbol{Q} \\
&\qquad \cdot [\mathrm{diag}(1,\lambda_2,\ldots,\lambda_{N-1})]^P\boldsymbol{Q}^T\boldsymbol{x}[0] \\
&\quad = (1/N)\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{x}[0] \qquad (16)
\end{aligned}$$

where the eigenvalue decomposition of $\boldsymbol{W}$ is given by $\boldsymbol{W} = \boldsymbol{Q}\mathrm{diag}(1,\lambda_2,\ldots,\lambda_{N-1})\boldsymbol{Q}^T$. Using the definition of line matrices and the fact that $\boldsymbol{x}[i] = \boldsymbol{W}^i\boldsymbol{x}[0]$, we arrive at

$$\begin{aligned}
&(1/N)\mathbf{1}_N\mathbf{1}_N^T\boldsymbol{x}[0] \\
&\quad = \boldsymbol{x}'[T] \\
&\quad = \boldsymbol{L}_{\beta_{T-1}}\cdots\boldsymbol{L}_{\beta_0}\boldsymbol{x}'[0] \\
&\quad = [\boldsymbol{I}-\beta_{T-1}(\boldsymbol{I}-\boldsymbol{W})]\cdots[\boldsymbol{I}-\beta_0(\boldsymbol{I}-\boldsymbol{W})]\boldsymbol{W}^P\boldsymbol{x}[0] \\
&\quad = \left(\sum_{l=0}^{T}\alpha_{T-l}\boldsymbol{W}^l\right)\boldsymbol{W}^P\boldsymbol{x}[0] \\
&\quad = \sum_{l=0}^{T}\alpha_{T-l}\boldsymbol{W}^{l+P}\boldsymbol{x}[0] \\
&\quad = \sum_{l=0}^{T}\alpha_{T-l}\boldsymbol{x}[l+P]
\end{aligned}$$

where the scalars $\alpha_0,\ldots,\alpha_T$ can be computed from $\beta_0,\ldots,\beta_{T-1}$ (in turn, the sequence β_i can be computed from the eigenvalues of $\boldsymbol{W}$). Equation (17) implies that $\boldsymbol{h} := [\alpha_0\ \cdots\ \alpha_T\ 0\ \cdots\ 0]^T \in \mathbb{R}^M$ satisfies $(1/N)\mathbf{1}_N^T\boldsymbol{x}[0] = \boldsymbol{h}^T\boldsymbol{y}_k[i]$ for every $k \in \mathcal{N}$, $i \geq M-1$, and initial condition $\boldsymbol{x}[0]$, which shows that $\mathcal{K}$ is nonempty. Note that $\boldsymbol{h}$ is not necessarily unique. Indeed, for sufficiently large M and i, both $[\alpha_0\ \cdots\ \alpha_T\ 0\ \cdots\ 0]^T \in \mathbb{R}^M$ and $[0\ \alpha_0\ \cdots\ \alpha_T\ 0\ \cdots\ 0]^T \in \mathbb{R}^M$ belong to $\mathcal{K}$.[10]

APPENDIX III
PROOF OF THEOREM 2

Proof:

a) For fixed $\boldsymbol{h}^\star \in \mathcal{K}$, the sequence $\|\boldsymbol{h}_k[i]-\boldsymbol{h}^\star\|$ is monotone nonincreasing (by assumption) and bounded below ($0 \leq \|\boldsymbol{h}_k[i]-\boldsymbol{h}^\star\|$), so it converges and $\|\boldsymbol{h}_k[i]\|$ is bounded above, i.e., there exists $B \in \mathbb{R}$ such that $\|\boldsymbol{h}_k[i]\| \leq B$. Define $\boldsymbol{y}^\star$ by $\boldsymbol{y}^\star := \lim_{i\to\infty}\boldsymbol{y}_k[i] = v^\star\mathbf{1}_M \in \mathbb{R}^M (\forall k \in \mathcal{N})$. Using the Cauchy–Schwartz inequality and the fact that $\boldsymbol{h}_k[i]^T\mathbf{1}_M = 1$ at any time instant, we deduce

$$\begin{aligned}
&|\boldsymbol{h}_k[i+1]^T\boldsymbol{y}_k[i] - v^\star| \\
&\quad = |\boldsymbol{h}_k[i+1]^T\boldsymbol{y}_k[i] - (\boldsymbol{h}_k[i+1]^T\mathbf{1}_M)v^\star| \\
&\quad = |\boldsymbol{h}_k[i+1]^T(\boldsymbol{y}_k[i]-\boldsymbol{y}^\star)| \\
&\quad \leq \|\boldsymbol{h}_k[i+1]\|\|\boldsymbol{y}_k[i]-\boldsymbol{y}^\star\| \\
&\quad \leq B\|\boldsymbol{y}_k[i]-\boldsymbol{y}^\star\| \to 0
\end{aligned}$$

as $i \to \infty$.

b) For $i \geq M-1$, we know that $v^\star = (\boldsymbol{h}^\star)^T\boldsymbol{y}[i]$. Therefore, by the Cauchy–Schwartz inequality,

$$\begin{aligned}
&|v_k[i] - v^\star| \\
&\quad = |\boldsymbol{h}_k[i+1]^T\boldsymbol{y}_k[i] - (\boldsymbol{h}^\star)^T\boldsymbol{y}_k[i]| \\
&\quad = |(\boldsymbol{h}_k[i+1]-\boldsymbol{h}^\star)^T\boldsymbol{y}_k[i]| \\
&\quad \leq \|\boldsymbol{h}_k[i+1]-\boldsymbol{h}^\star\|\,\|\boldsymbol{y}_k[i]\| \\
&\quad \leq S\|\boldsymbol{h}_k[i+1]-\boldsymbol{h}^\star\|,
\end{aligned}$$

which is valid for every $\boldsymbol{h}^\star \in \mathcal{K}$. In particular, at time i, choose $\boldsymbol{h}^\star = P_{\mathcal{K}}(\boldsymbol{h}_k[i+1])$ and the proof is complete.

[10] We can easily show that the $\mathcal{K}$ is convex as follows. Let $\boldsymbol{h}_{(1)}$ and $\boldsymbol{h}_{(2)}$ be any two vectors belonging to $\mathcal{K}$. Define $\boldsymbol{h}_{(3)} \in \mathbb{R}^M$ by $\boldsymbol{h}_{(3)} := \nu\boldsymbol{h}_{(1)} + (1-\nu)\boldsymbol{h}_{(2)}$, where $\nu \in [0,1]$. Then $\boldsymbol{h}_{(3)}^T\boldsymbol{y}_k[i] = \nu\boldsymbol{h}_{(1)}^T\boldsymbol{y}_k[i] + (1-\nu)\boldsymbol{h}_{(2)}^T\boldsymbol{y}_k[i] = (1/N)\mathbf{1}_N^T\boldsymbol{x}[0]$ for every $\nu \in [0,1]$, which shows that $\boldsymbol{h}_{(3)} \in \mathcal{K}$, and thus $\mathcal{K}$ is convex.

REFERENCES

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

[2] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[3] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65–78, Sep. 2004.

[4] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad hoc networks," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2004, pp. 11–19.

[5] R. Olfati-Saber, "Ultrafast consensus in small-world networks," in *Proc. Amer. Control Conf.*, Jun. 2005, vol. 4, pp. 2371–2378.

[6] D. Scherber and H. C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 776–787, Apr. 2005.

[7] S. Sundaram and C. N. Hadjicostis, "Finite-time distributed consensus in graphs with time-invariant topologies," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2007, pp. 711–716.

[8] L. Xiao, S. P. Boyd, and S. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007.

[9] E. Kokiopoulou and P. Frossard, "Accelerating distributed consensus using extrapolation," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 665–668, Oct. 2007.

[10] L. Pescosolido, S. Barbarossa, and G. Scutari, "Radar sensor networks with distributed detection capabilities," in *Proc. IEEE Radar Conf. 2008*, Rome, Italy, May 26–30, 2008.

[11] B. N. Oreshkin, T. C. Aysal, and M. J. Coates, "Distributed average consensus with increased convergence rate," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar./Apr. 2008, pp. 2285–2288.

[12] D. Ustebay, M. Coates, and M. Rabbat, "Greedy gossip with eavesdropping," in *Proc. 3rd Int. Symp. Wireless Pervasive Comput. (ISWPC)*, May 2008, pp. 759–763.

[13] D. Ustebay, B. Oreshkin, M. Coates, and M. Rabbat, "Rates of convergence for greedy gossip with eavesdropping," in *Proc. 46th Annu. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, Sep. 2008, pp. 367–374.

[14] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, Jul. 2008.

[15] E. Kokiopoulou and P. Frossard, "Polynomial filtering for fast convergence in distributed consensus," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 342–354, Jan. 2009.
[16] T. C. Aysal, B. N. Oreshkin, and M. J. Coates, "Accelerated distributed average consensus via localized node state prediction," *IEEE Trans. Signal Process.*, vol. 57, no. 4, pp. 1563–1576, Apr. 2009.
[17] K. Jbilou and H. Sadok, "Vector extrapolation methods. Applications and numerical comparison," *J. Comput. Appl. Math.*, vol. 122, no. 1–2, pp. 149–165, Oct. 2000.
[18] I. Yamada, "Adaptive projected subgradient method: A unified view for projection based adaptive algorithms," *J. IEICE*, vol. 86, no. 8, pp. 654–658, Aug. 2003, in Japanese.
[19] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numer. Funct. Anal. Optim.*, vol. 25, no. 7/8, pp. 593–617, 2004.
[20] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fundam.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.
[21] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.
[22] H. Stark and Y. Yang, *Vector Space Projections—A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. New York: Wiley, 1998.
[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[24] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
[25] T. Hinamoto and S. Maekawa, "Extended theory of learning identification," *Trans. IEE Jpn.*, vol. 95, no. 10, pp. 227–234, 1975.
[26] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *IEICE Trans.*, vol. 67-A, no. 5, pp. 126–132, Feb. 1984.
[27] I. Yamada, K. Slavakis, and K. Yamada, "An efficient robust adaptive filtering algorithm based on parallel subgradient projection techniques," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1091–1101, May 2002.
[28] A. H. Sayed, *Fundamentals of Adaptive Filtering*. Hoboken, NJ: Wiley, 2003.
[29] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[30] G. Golub and W. Kahan, "Calculating the singular values and the pseudo-inverse of a matrix," *SIAM J. Numer. Anal.*, ser. B, vol. 2, no. 2, pp. 205–224, 1965.
[31] M. Grand and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming (web page and software). Oct. 2008 [Online]. Available: http://stanford.edu/~boyd/cvx
[32] M. Grand and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control (a Tribute to M. Vidyasagar), Lecture Notes in Control and Information Sciences*, V. Blondel, S. Boyd, and H. Kimura, Eds. New York: Springer, Oct. 2008, pp. 95–110.
[33] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

Renato L. G. Cavalcante (M'08) received the Electronics Engineering degree from the Instituto Tecnológico de Aeronáutica (ITA), Brazil, in 2002, and the M.E. and Ph.D. degrees in communications and integrated systems from the Tokyo Institute of Technology, Japan, in 2006 and 2008, respectively.

From April 2003 to April 2008, he was a recipient of a Japanese Government (MEXT) Scholarship. From 2008 to 2009, he was a Research Associate with The University of Edinburgh, U.K. He is currently a Research Fellow with the University of Southampton, U.K. His current interests are in signal processing for distributed systems and wireless communications.

Dr. Cavalcante received the Excellent Paper Award from the IEICE in 2006 and the IEEE Signal Processing Society (Japan Chapter) Student Paper Award in 2008.

Bernard Mulgrew (M'88–SM'07) received the B.Sc. degree from Queen's University Belfast in 1979 and the Ph.D. degree from the University of Edinburgh, U.K. in 1987.

After graduation, he was a Development Engineer for four years in the Radar Systems Department at Ferranti, Edinburgh, U.K. From 1983 to 1986, he was a Research Associate in the Department of Electronics and Electrical Engineering at the University of Edinburgh, studying the performance and design of adaptive filter algorithms. In 1987, he was appointed to a lectureship at the University of Edinburgh, where he currently holds the SELEX S&AS/Royal Academy of Engineering Research Chair in Signal Processing and is Head of the Institute for Digital Communications at the University of Edinburgh. His research interests are in adaptive signal processing and estimation theory and in their application to communications, radar and audio systems. He is a coauthor of three books on signal processing and over 70 journal papers.

Dr. Mulgrew is a fellow of the Royal Academy of Engineering, a fellow of the Royal Society of Edinburgh, and a fellow of the IET.