INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2014; **27**:3266–3279
Published online 1 April 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2540

Filtering approaches to accelerated consensus in diffusion sensor networks

Emad Abd-Elrady[1,*,†] and Bernard Mulgrew[2]

[1]*Higher Colleges of Technology, Abu Dhabi Women's College (ADWC), 41012 Abu Dhabi, UAE*
[2]*Institute for Digital Communication, School of Engineering, The University of Edinburgh, EH9 3JL Edinburgh, UK*

SUMMARY

The main objective in distributed sensor networks is to reach agreement or consensus on values acquired by the sensors. A common methodology to approach this problem is using the iterative and weighted linear combination of those values to which each sensor has access. Different methods to compute appropriate weights have been extensively studied, but the resulting iterative algorithm still requires many iterations to provide a fairly good estimate of the consensus value. In this paper, different accelerating consensus approaches based on adaptive and non-adaptive filtering techniques are studied and applied on the problem of acoustic source localization using the adaptive projected subgradient method. A comparative simulation study shows that the non-adaptive polynomial filters based on Newton's interpolating polynomials and semi-definite programming can provide more accelerated consensus and better estimation accuracy than adaptive filters evaluated using constrained affine projection algorithm or stochastic gradient algorithm provided that the network topology is known beforehand. Copyright © 2013 John Wiley & Sons, Ltd.

Received 28 December 2011; Revised 9 December 2012; Accepted 23 February 2013

KEY WORDS: adaptive filtering; adaptive projected subgradient method; consensus; diffusion networks; distributed algorithms; optimization methods; semi-definite programming

1. INTRODUCTION

Distributed sensor network consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, or motion [1–7]. The development of sensor networks was motivated by military applications such as battlefield surveillance and are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment monitoring, healthcare applications, home automation, and traffic control.

Signal processing and adaptive learning can operate in three basic modes across such sensor networks: *global*, *incremental*, and *diffusion*. In the global mode, the data from all the nodes are sent to a single fusion center that forms an estimate of the parameters of interest. The requirement for measurement data from all the nodes places a significant burden on the available communication bandwidth and implies that the fusion center is distinctly different from all the other nodes in terms of processing power and communications capability. In the incremental mode, the nodes are connected in a cyclic pattern—a node combines its local data with the estimate received from the previous node and sends the new estimate to the next node in the network. This mode of cooperation requires the minimum amount of power and is easy to implement in a network with a fairly small number of nodes. However, defining a path through the network may not be practical for large

*Correspondence to: Emad Abd-Elrady, Abu Dhabi Women's College, Higher Colleges of Technology, P.O. Box 41012 Abu Dhabi, UAE.
†E-mail: emad.abdelrady@hct.ac.ae

networks (or for dynamic re-configuration), learning rates are relatively slow because of the cyclic operation and a break in the path may have a catastrophic effect.

In the diffusion mode, each node uses its own local information and a consensus of the estimates from a subset of its neighbors to form its own estimate. Each node iteratively repeats this operation as communication opportunities and available data permit. A single node may only have direct access to local data, and the estimates from a small number of its neighbors but its own estimate can converge over time through the diffusion processes to the same quality estimate as that of a fusion center in the global mode. The diffusion mode provides faster learning rates than the incremental mode and can deal with a large number of sensors, node failures, changing topologies, and/or communication problems between the nodes. However, the diffusion mode also presents additional challenges in both the development and analysis of the signal processing algorithms due to the incorporation of this consensus concept. It also raises new questions in selecting an optimal consensus strategy for particular applications and in seeking to accelerate consensus.

Recently, the problem of accelerating consensus in diffusion sensor networks attracts the attention of many researchers (see [8–10]). The existing approaches to accelerate the consensus is based on making use of the memory of the sensing node by saving a sequence of consecutive past local estimates. Then, developing adaptive or fixed filter to filter these past estimates and using the output of the filter as the current local estimate that is used for communication with other nodes. The design of an adaptive filter can be performed using constrained affine projection algorithm (CAPA) [10] or stochastic gradient algorithm (SGA) as shown in this paper. Also, a non-adaptive polynomial filter can be designed as shown in [8, 9] using Newton interpolation polynomials (NIPs) or semi-definite programming (SDP).

This paper is organized as follows. In Sections 2 and 3, the basic concepts of convex analysis and the adaptive projected subgradient method (APSM) are reviewed, respectively. The problem of acoustic source localization (ASL) is discussed in Section 4. In Section 5, the adaptive filtering approach for accelerating consensus in the ASL problem is presented. The non-adaptive polynomial filtering approach is introduced in Section 6. In Section 7, a comparative simulation study between different approaches for accelerating consensus in ASL problem is given. Section 8 comes to conclusions.

2. PRELIMINARIES

2.1. Basic tools 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 $\|\boldsymbol{X}\|_2 := \max \left\{ \sqrt{\lambda} \mid \lambda \text{ is an eigenvalue of} \boldsymbol{X}^T \boldsymbol{X} \right\}$, which satisfies $\|\boldsymbol{X}\boldsymbol{y}\| \leqslant \|\boldsymbol{X}\|_2 \|\boldsymbol{y}\|$ for any vector $\boldsymbol{y}$ of compatible size [11].

A set is said to be convex if $\boldsymbol{v} = \alpha \boldsymbol{v}_1 + (1-\alpha)\boldsymbol{v}_2 \in C \subset \mathbb{R}^N$ for every $\boldsymbol{v}_1, \boldsymbol{v}_2 \in C$ and $0 \leqslant \alpha \leqslant 1$ [12, 13]. 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 unique 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 $\forall \nu \in [0, 1]$, $\theta(\nu \boldsymbol{x} + (1-\nu)\boldsymbol{y}) \leqslant \nu\theta(\boldsymbol{x}) + (1-\nu)\theta(\boldsymbol{y})$ [13]. Let θ be a continuous convex function. The subdifferential of θ at $\boldsymbol{y}$ is the set of all the subgradients of θ at $\boldsymbol{y}$ [14]:

$$\partial\theta(\boldsymbol{y}) := \left\{ \boldsymbol{a} \in \mathbb{R}^N \mid \theta(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{a} \rangle \leqslant \theta(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^N \right\} \neq \emptyset \tag{1}$$

2.2. Distributed average consensus

Consider a network with N nodes, where at time $i = 0$, each node has an estimate $x_k[0] \in \mathbb{R}$ of some parameter of interest, $k \in \mathcal{N}$, and $\mathcal{N} := 1, \cdots, N$ is the node set. Define $\mathcal{G} := (\mathcal{N}, \mathcal{E})$ as the undirected connected graph with node set $\mathcal{N}$ and edge set $\mathcal{E}$, where $\{l, k\} \in \mathcal{E}$ is an unordered pair of distinct connected nodes. The set of neighbors of node k is denoted by $\mathcal{N}_k := \{l | \{l, k\} \in \mathcal{E}\}$. In many applications, the objective is to develop an algorithm such that each node converges to the

DOI: 10.1002/dac

same estimate $x^* := (1/N) \sum_{k \in \mathcal{N}} x_k[0]$ or the average in case of using a sensor fusion center in global mode. In diffusion mode, this should be achieved only on the basis of local communication between nodes.

The following iterative and weighted linear algorithms are commonly used to reach consensus:

$$x_k[i+1] = w_{k,k} x_k[i] + \sum_{l \in \mathcal{N}_k} w_{k,l} \left(x_l[i] - x_k[i] \right) \tag{2}$$

where $k \in \mathcal{N}$ and $w_{k,l}$ is a weight associated with the edge $\{k, l\}$. Setting $w_{k,l} = 0$ for $l \notin \mathcal{N}_k$ and $w_{k,k} = 1 - \sum_{j \in \mathcal{N}_k} w_{k,j}$, (2) can be written in a more compact form as

$$\boldsymbol{x}[i+1] = \boldsymbol{W} \boldsymbol{x} \tag{3}$$

where $\boldsymbol{x}[i] = (x_1[i] \; x_2[i] \; \cdots \; x_N[i])^T \in \mathbb{R}^N$ with initial condition $\boldsymbol{x}[0] = (x_1[0] \; x_2[0] \; \cdots \; x_N[0])^T \in \mathbb{R}^N$, $\boldsymbol{W}$ is the consensus matrix [15] and the component of the kth row and lth column of $\boldsymbol{W} \in \mathbb{R}^{N \times N}$ is $w_{k,l}$.

3. THE ADAPTIVE PROJECTED SUBGRADIENT METHOD

Consider a dynamic network with a similar topology to the one described in 2.2. At time i, let $\mathcal{G} := (\mathcal{N}, \mathcal{E}[i])$ be the possibly time-varying undirected connected graph of the network and assume that $\Theta_k[i] : \mathbb{R}^M \to [0, \infty), \forall i \in \mathbb{N}$ is a sequence of continuous convex cost functions known by the kth node. In addition, assume that at time i, node k has an estimate $\boldsymbol{h}_k[i] \in \mathbb{R}^M$ of the minimizer of $\Theta_k[i]$ and access to the estimates of its neighbors, the estimates $\boldsymbol{h}_l[i], l \in \mathcal{N}_k := \{l | \{l, k\} \in \mathcal{E}[i]\}$. At time i, the function $\Theta_k[i]$ is the local cost function for node k, and the target cost function $\Theta[i] : \mathbb{R}^M \to [0, \infty)$ as the sum of the N time-varying local cost functions, that is,

$$\Theta[i](\boldsymbol{h}) = \sum_{k \in \mathcal{N}} \Theta_k[i](\boldsymbol{h}) \tag{4}$$

Let us define for each node k a convex set Ω_k that contains the estimate $\boldsymbol{h}_k$ of the minimizer of $\Theta_k[i]$. Also, assume that

$$\Omega[i] := \bigcap_{k \in \mathcal{N}} \Omega_k[i] \neq \emptyset \tag{5}$$

where

$$\Omega_k[i] := \left\{ \boldsymbol{h} \in \mathbb{R}^M | \Theta_k[i](\boldsymbol{h}) = \Theta_k^*[i] := \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta_k[i](\boldsymbol{h}) \right\}, \; k \in \mathcal{N} \tag{6}$$

Therefore, any $\boldsymbol{h}^*[i] \in \Omega[i]$ is a minimizer of (4). The objective is to minimize (asymptotically) the target cost function in (4) in every node. In addition, the nodes should agree with the minimizer, that is, $\boldsymbol{h}_k[i] = \boldsymbol{h}^*[i]$ for some $\boldsymbol{h}^*[i]$, regardless the fact that each node has only partial knowledge of $\Theta[i](\boldsymbol{h})$.

The adaptive projected subgradient method (APSM) was introduced in [16] and applied for diffusion mode networks in [15]. Also, different properties of the APSM such as monotone approximation, boundedness, asymptotic optimality, and convergence to consensus are discussed in details in [10, 15, 16]. Let

$$\boldsymbol{\psi}[i] := \left(\boldsymbol{h}_1[i]^T \quad \boldsymbol{h}_2[i]^T \quad \cdots \quad \boldsymbol{h}_N[i]^T \right)^T \tag{7}$$

where $\boldsymbol{h}_k[i]$ is the estimate of $\boldsymbol{h}_k^*[i] \in \Omega_k[i]$ in the kth node. The APS algorithm follows as [10, 15–17]

$$\boldsymbol{\psi}[i+1] = \boldsymbol{P}[i] \left(\boldsymbol{\psi}[i] - \begin{bmatrix} \mu_1[i]\eta_1[i]\Theta_1'[i](\boldsymbol{h}_1[i]) \\ \vdots \\ \mu_N[i]\eta_N[i]\Theta_N'[i](\boldsymbol{h}_N[i]) \end{bmatrix} \right) \tag{8}$$

where $\mu_k[i] \in [0, 2]$, $\eta_k[i] = \frac{\Theta_k[i](\boldsymbol{h}[i]) - \Theta_k^*[i]}{\|\Theta'_k[i](\boldsymbol{h}_k[i])\|^2 + \delta_k[i]}$, $\delta_k[i] > 0$ is an arbitrary small (bounded) number, $\Theta_k^*[i] := \inf_{\boldsymbol{h} \in \mathbb{R}^M} \Theta_k[i](\boldsymbol{h})(k \in \mathcal{N})$, and $\Theta'_k[i](\boldsymbol{h}_k[i]) \in \partial\Theta_k[i](\boldsymbol{h}[i])$. Here, $\boldsymbol{P} \in \mathbb{R}^{MN \times MN}$ is an extended consensus matrix satisfying $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}$ and $\boldsymbol{P}^T\boldsymbol{x} = \boldsymbol{x}$ for every vector $\boldsymbol{x} \in C := \{[\boldsymbol{a}^T \cdots \boldsymbol{a}^T]^T \in \mathbb{R}^{MN} | \boldsymbol{a} \in \mathbb{R}^M\}$, and the m largest singular values of $\boldsymbol{P}$ are equal to one and the remaining $MN - M$ singular values are strictly less than one. Also, the extended consensus matrix $\boldsymbol{P}$ has some other properties that listed in [15, 17] and quoted here as Lemma 1.

Lemma 1
Let $\boldsymbol{P} \in \mathbb{R}^{MN \times MN}$ be an extended consensus matrix, $\boldsymbol{e}_k \in \mathbb{R}^M$ be the vector of zeros, except for its kth entry, which is set to one, and C be the subspace $C := \text{span}\{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_M\}$, where $\boldsymbol{b}_k = (\boldsymbol{1}_N \otimes \boldsymbol{e}_k)/\sqrt{N} \in \mathbb{R}^{MN}, \boldsymbol{1}_N \in \mathbb{R}^N$ is the vector of ones, and $\otimes$ denotes the Kronecker product. At time i, define $\boldsymbol{\psi}[i] := \left(\boldsymbol{h}_1[i]^T \ \boldsymbol{h}_2[i]^T \ \cdots \ \boldsymbol{h}_N[i]^T\right)^T$. Then the following properties hold:

(1) Any consensus matrix $\boldsymbol{P}$ can be decomposed into $\boldsymbol{P} = \boldsymbol{B}\boldsymbol{B}^T + \boldsymbol{X}$, where $\boldsymbol{B} := (\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_M) \in \mathbb{R}^{MN \times M}$, and $\boldsymbol{X} \in \mathbb{R}^{MN \times MN}$ is a matrix satisfying $\boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{X} = \boldsymbol{0}$ and $\|\boldsymbol{X}\|_2 < 1$. Note that $\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{1}_M$ by construction.
(2) The nodes of the network are in agreement at time i, that is, $\boldsymbol{h}_1[i] = \cdots = \boldsymbol{h}_N[i]$ or equivalently $\boldsymbol{\psi}[i] \in C$, if and only if $\boldsymbol{\psi}[i] \in \text{null}\left(\boldsymbol{I}_{MN} - \boldsymbol{B}\boldsymbol{B}^T\right)$ or $\left(\boldsymbol{I}_{MN} - \boldsymbol{B}\boldsymbol{B}^T\right)\boldsymbol{\psi}[i] = \boldsymbol{0}$ where $\boldsymbol{I}_{MN} \in \mathbb{R}^{MN \times MN}$ is the identity matrix.

4. THE ACOUSTIC SOURCE LOCALIZATION PROBLEM

The problem of ASL as intersection of closed convex sets is considered in this paper. Consider a sensor network with node set $\mathcal{N}$, where the sensors are distributed at known spatial locations denoted by $\boldsymbol{r}_k \in \mathbb{R}^2$ (extension to $\mathbb{R}^3$ is straightforward), $k \in \mathcal{N}$. If there exists an acoustic source located at the unknown position $\boldsymbol{h}^* \in \mathbb{R}^2$, a model for the estimated signal strength at the node k is [18]

$$y_k = \frac{A}{\|\boldsymbol{r}_k - \boldsymbol{h}^*\|} + v_k \in \mathbb{R} \tag{9}$$

where A is the energy of the acoustic signal and v_k is a zero-mean white Gaussian noise with variance σ_{v_k}.

Two approaches were considered in [17] to locate the acoustic source. The first approach ignores the presence of noise and finds the source location $\boldsymbol{h}^*$ in the intersection of *fixed* radius disks given by

$$D_k := \left\{\boldsymbol{h} \in \mathbb{R}^2 | \|\boldsymbol{h} - r_k\| \leqslant \sqrt{A/y_k}\right\} (k \in \mathcal{N}) \tag{10}$$

Therefore, a reasonable estimate of $\boldsymbol{h}^*$ is

$$\widehat{\boldsymbol{h}}^* \in D = \bigcap_{k \in \mathcal{N}} D_k, \tag{11}$$

which can be found also using the projection onto convex sets algorithm introduced in [18].

The second approach of [17] considers the presence of noise and trying to compensate for the shrinkage in the disk radius due to its existence by proposing disks with *time-varying* radius given by

$$C_k[i] := \left\{\boldsymbol{h} \in \mathbb{R}^2 | \|\boldsymbol{h} - r_k\| \leqslant \sqrt{\frac{A}{y_k - e_k[i]}}\right\} \forall (y_k - e_k[i]) > 0 \quad (C_k[i] := \mathbb{R}^2 \text{ otherwise}) \tag{12}$$

where $e_k[i] > 0$ is an inflation parameter for the radius of $C_k[i]$.

The following (possibly time-varying) local cost function was used in [17] to devise a fast algorithm

$$\Theta_k[i](\boldsymbol{h}) = \sum_{l \in \mathcal{N}_k} \kappa_{l,k} d^2(\boldsymbol{h}, C_l[i]) \tag{13}$$

where $\kappa_{l,k} = y_k / \left(\sum_{l \in \mathcal{N}_k} y_l \right)$ is the weight that node k assigns to the disk of its neighbor $l \in \mathcal{N}_k$ and $d(\boldsymbol{h}, C_l[i]) = \|\boldsymbol{h} - P_{C_l[i]}(\boldsymbol{h})\|$. The weights should be non-negative and satisfy $\sum_{l \in \mathcal{N}_k} \kappa_{l,k} = 1$ for every $k \in \mathcal{N}$. Assuming that $\bigcap_{k \in \mathcal{N}} C_k[i] \neq 0$, $\widehat{\boldsymbol{h}}^* \in \bigcap_{k \in \mathcal{N}_k} C_k[i]$ is a minimizer of every local cost function $\Theta_k[i](\boldsymbol{h})$. Hence, $\widehat{\boldsymbol{h}}^*$ is also a minimizer of the global cost function, that is, $\Theta[i](\widehat{\boldsymbol{h}}^*) = \sum_{k \in \mathcal{N}} \Theta_k[i](\widehat{\boldsymbol{h}}^*) = 0$. In this case, the APSA in (8) takes the form in Table I.

5. ADAPTIVE FILTERING APPROACH FOR ACCELERATING CONSENSUS

It is shown in [10] that the consensus value can be computed in a finite time by filtering the estimate of each sensor, $x_k[i] \forall k \in \mathcal{N}$, obtained with (3). Therefore, by defining

$$\boldsymbol{y}_k[i] := \begin{pmatrix} x_k[i] & x_k[i-1] & \cdots & x_k[i-M+1] \end{pmatrix}^T \in \mathbb{R}^M \quad (i \geqslant M-1) \tag{14}$$

The *nonempty* set of linear filters $\boldsymbol{f}$ that can compute the consensus value $x^* := (1/N) \sum_{k \in \mathcal{N}} x_k[0]$ for every initial node condition $x_k[0]$ by filtering any m consecutive samples of the local information $x_k[i]$ is defined as

$$\mathcal{K} := \left\{ \boldsymbol{f} \in \mathbb{R}^M \,|\, \boldsymbol{f}^T \boldsymbol{y}_k[i] = x^* \ \forall k \in \mathcal{N} \right\} \tag{15}$$

Table I. Algorithm I: acoustic source localization using adaptive projected subgradient method.

for time $i = 0, 1, \cdots$

for $k = 1 : N$

Choose $\delta_k > 0$, $\varepsilon = \sqrt{A/(y_k - e_k[i])}$ and $\mu_k[i] \in (0, \mathcal{M}_k[i])$ where

$$\mathcal{M}_k[i] = \begin{cases} 1 & \text{if } \boldsymbol{h}_k[i] \in \bigcap_{l \in \mathcal{N}_k} C_l[i] \\ \dfrac{\sum_{l \in \mathcal{N}_k} \|\kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2}{\|\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2 + \delta_k[i]} & \text{otherwise} \end{cases}$$

$$P_{C_l[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in C_l[i] \\ \boldsymbol{r}_k + \frac{\varepsilon}{\|\boldsymbol{h} - \boldsymbol{r}_k\|}(\boldsymbol{h} - \boldsymbol{r}_k) & \text{otherwise} \end{cases}$$

$$\boldsymbol{h}'_k[i] = \boldsymbol{h}_k[i] + \mu_k[i] \left(\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i] \right)$$

end

$$\begin{bmatrix} \boldsymbol{h}_1[i+1] \\ \vdots \\ \boldsymbol{h}_N[i+1] \end{bmatrix} = \boldsymbol{P}[i] \begin{bmatrix} \boldsymbol{h}'_1[i+1] \\ \vdots \\ \boldsymbol{h}'_N[i+1] \end{bmatrix}$$

end

DOI: 10.1002/dac

The suggested constrained affine projection algorithm (CAPA) in [10] is based on set-theoretic adaptive filters [19, 20]. The CAPA defines the following hyperplanes:

$$\mathcal{C} := \left\{ \boldsymbol{f} \in \mathbb{R}^M \,|\, \boldsymbol{f}^T \mathbf{1}_M = 1 \right\} \tag{16}$$

$$\mathcal{D}_k[i,n] := \left\{ \boldsymbol{f} \in \mathbb{R}^M \,|\, \boldsymbol{f}^T \boldsymbol{\delta}_k[i,n] = 0 \right\} \supseteq \mathcal{K} \tag{17}$$

where $\mathbf{1}_M$ is a $M \times 1$ vector of ones and the normalized affine projection $\boldsymbol{\delta}_k[i,n], \forall i, n \geqslant M-1$ is defined as

$$\boldsymbol{\delta}_k[i,n] := \frac{(\boldsymbol{y}_k[i] - \boldsymbol{y}_k[n])M}{\|\boldsymbol{y}_k[i] - \boldsymbol{y}_k[n]\| + \epsilon}, \quad \epsilon > 0 \quad \text{is arbitrarily small} \tag{18}$$

The CAPA estimates the adaptive filter $\boldsymbol{f}$ that belongs to the set $\mathcal{V}_k[i]$, which combines sets $\mathcal{C}$ and $\mathcal{D}_k$. The set $\mathcal{V}_k[i]$ is defined as [10]

$$\mathcal{V}_k[i] := \left\{ \boldsymbol{f} \in \mathbb{R}^M \,|\, \boldsymbol{H}_k[i]^T \boldsymbol{f} = [0 \ \cdots \ 0 \ 1]^T \right\} = \bigcap_{j=1}^{Q[i]} \mathcal{D}_k[i, i-j] \cap \mathcal{C} \tag{19}$$

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

The CAPA algorithm for estimating the adaptive filter $\boldsymbol{f}$ follows as:

$$\boldsymbol{f}_k[i+1] = P_{\mathcal{C}} \left(\boldsymbol{f}_k[i] + \gamma_k[i] (P_{\mathcal{V}_k[i]}(\boldsymbol{f}_k[i]) - \boldsymbol{f}_k[i]) \right) \tag{20}$$

where $\gamma_k[i] \in (0,2)$ is the step size, $P_{\mathcal{C}}(\boldsymbol{f}) = \boldsymbol{f} - (\boldsymbol{f}^T \mathbf{1}_M - 1)\mathbf{1}_M / \|\mathbf{1}_M\|^2$ is the projection of $\boldsymbol{f}$ onto $\mathcal{C}$, and

$$P_{\mathcal{V}_k[i]}(\boldsymbol{f}_k[i]) = \boldsymbol{f}_k[i] - \left(\boldsymbol{H}_k[i]^{\dagger} \right)^T \left(\boldsymbol{H}_k[i]^T \boldsymbol{f}_k[i] - [0 \ \cdots \ 1]^T \right) \tag{21}$$

is the projection of $\boldsymbol{f}_k[i]$ onto $\mathcal{V}_k[i]$ and $(.)^{\dagger}$ denotes the Moore–Penrose pseudoinverse [21].

The non-attractive point in (21) is the need for the pseudo-inverse which may cause numerical problems and involves more computational complexity. For this reason, the stochastic gradient algorithm [22, 23] can be used in this paper to avoid the use of the pseudoinverse. The stochastic gradient algorithm (SGA) can be easily deduced by constructing the error as

$$\boldsymbol{e}_k[i] = \boldsymbol{H}_k[i]^T \boldsymbol{f}_k[i] - [0 \ \cdots \ 0 \ 1]^T \tag{22}$$

and follows similar to (20) as

$$\boldsymbol{f}_k[i+1] = P_{\mathcal{C}} \left(\boldsymbol{f}_k[i] - \alpha_k[i] \boldsymbol{\Delta}_k[i] \right) \tag{23}$$

where $\alpha_k[i]$ is the step size and $\boldsymbol{\Delta}_k$ is the gradient evaluated as

$$\boldsymbol{\Delta}_k[i] = \frac{\mathrm{d}\left(\boldsymbol{e}_k[i]^T \boldsymbol{e}_k[i] \right)}{\mathrm{d}\boldsymbol{f}_k} = 2\boldsymbol{H}_k[i]\boldsymbol{e}_k[i] \tag{24}$$

Remark 1

As mentioned in [10], there is no guarantee using the CAPA to improve the estimate of the consensus.

The suggested algorithms in this paper for ASL using the CAPA and SGA are denoted as algorithms II and III and shown in Tables II and III; respectively.

Remark 2

As it is clear from Tables II and III, the adaptive filters designed using the CAPA and SGA are applied on the node estimates evaluated through the compact averaging formula using the extended consensus matrix. The output filtered estimates are denoted as $\boldsymbol{h}_k^f[i]$.

Table II. Algorithm II: acoustic source localization using adaptive projected subgradient method and constrained affine projection algorithm.

for time $i = 0, 1, \cdots$

for $k = 1 : N$. Choose $\delta_k > 0$, $\varepsilon = \sqrt{A/(y_k - e_k[i])}$, $\gamma_k[i] \in (0, 2)$, $\boldsymbol{f}_k[0] = [1\ 0\ \cdots\ 0]^T$,

the memory number J, $Q[i] = \min(i - M, J)$, $\epsilon > 0$, and $\mu_k[i] \in (0, \mathcal{M}_k[i])$ where

$$\mathcal{M}_k[i] = \begin{cases} 1 & \text{if } \boldsymbol{h}_k[i] \in \bigcap_{l \in \mathcal{N}_k} C_l[i] \\ \frac{\sum_{l \in \mathcal{N}_k} \|\kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2}{\|\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2 + \delta_k[i]} & \text{otherwise} \end{cases}$$

$$P_{C_l[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in C_l[i] \\ \boldsymbol{r}_k + \frac{\varepsilon}{\|\boldsymbol{h} - \boldsymbol{r}_k\|}(\boldsymbol{h} - \boldsymbol{r}_k) & \text{otherwise} \end{cases}$$

$$\boldsymbol{h}'_k[i] = \boldsymbol{h}_k[i] + \mu_k[i] \left(\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i] \right)$$

for all $i, n \geq M$

$$\boldsymbol{y}_k[i] := \begin{pmatrix} \boldsymbol{h}'_k[i] & \boldsymbol{h}'_k[i-1] & \cdots & \boldsymbol{h}'_k[i-M+1] \end{pmatrix}^T$$

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

$$\boldsymbol{H}_k[i] := [\boldsymbol{\delta}_k[i, i-1]\ \boldsymbol{\delta}_k[i, i-2]\ \cdots\ \boldsymbol{\delta}_k[i, i - Q[i]]\ \mathbf{1}_M]$$

$$P_{\mathcal{V}_k[i]}(\boldsymbol{f}_k[i]) = \boldsymbol{f}_k[i] - (\boldsymbol{H}_k[i]^\dagger)^T (\boldsymbol{H}_k[i]^T \boldsymbol{f}_k[i] - [0\ \cdots\ 1]^T)$$

$$P_{\mathcal{C}}(\boldsymbol{f}_k[i]) = \boldsymbol{f}_{\boldsymbol{k}}[\boldsymbol{i}] - (\boldsymbol{f}_k[i]^T \mathbf{1}_M - 1)\mathbf{1}_M / \|\mathbf{1}_M\|^2$$

$$\boldsymbol{f}_k[i+1] = P_{\mathcal{C}}\left(\boldsymbol{f}_k[i] + \gamma_k[i](P_{\mathcal{V}_k[i]}(\boldsymbol{f}_k[i]) - \boldsymbol{f}_k[i])\right)$$

end

end

$$\begin{bmatrix} \boldsymbol{h}_1[i+1] \\ \vdots \\ \boldsymbol{h}_N[i+1] \end{bmatrix} = \boldsymbol{P}[i] \begin{bmatrix} \boldsymbol{h}'_1[i] \\ \vdots \\ \boldsymbol{h}'_N[i] \end{bmatrix}, \qquad \begin{bmatrix} \boldsymbol{h}^f_1[i+1] \\ \vdots \\ \boldsymbol{h}^f_N[i+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1[i+1]^T \boldsymbol{y}_1[i+1] \\ \vdots \\ \boldsymbol{f}_N[i+1]^T \boldsymbol{y}_N[i+1] \end{bmatrix}$$

end

6. NON-ADAPTIVE FILTERING APPROACH FOR ACCELERATING CONSENSUS

Non-adaptive polynomial filtering techniques were discussed in [8, 9] to exploit the memory of sensors or the values of previous estimates. The basic idea is to assume that the network topology is known a priori and find the fixed filter that shape the spectrum of the consensus matrix $\boldsymbol{W}$ in (3) to impact the magnitude of the second eigenvalue that mainly drives the consensus speed of the network. The authors in [9] presents two techniques for designing the filter. The first approach is

DOI: 10.1002/dac

Table III. Algorithm III: acoustic source localization using adaptive projected subgradient method and stochastic gradient algorithm.

for time $i = 0, 1, \cdots$

  for $k = 1 : N$. Choose $\delta_k > 0$, $\varepsilon = \sqrt{A/(y_k - e_k[i])}$, $\alpha_k[i]$, $\boldsymbol{f}_k[0] = [1\ 0\ \cdots\ 0]^T$

  and $\mu_k[i] \in (0, \mathcal{M}_k[i])$ where

$$\mathcal{M}_k[i] = \begin{cases} 1 & \text{if } \boldsymbol{h}_k[i] \in \bigcap_{l \in \mathcal{N}_k} C_l[i] \\ \frac{\sum_{l \in \mathcal{N}_k} \|\kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2}{\|\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2 + \delta_k[i]} & \text{otherwise} \end{cases}$$

$$P_{C_l[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in C_l[i] \\ \boldsymbol{r}_k + \frac{\varepsilon}{\|\boldsymbol{h} - \boldsymbol{r}_k\|}(\boldsymbol{h} - \boldsymbol{r}_k) & \text{otherwise} \end{cases}$$

$$\boldsymbol{h}'_k[i] = \boldsymbol{h}_k[i] + \mu_k[i] \left(\sum_{l \in \mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i] \right)$$

    for all $i, n \geq M$

$$\boldsymbol{y}_k[i] := \begin{pmatrix} \boldsymbol{h}'_k[i] & \boldsymbol{h}'_k[i-1] & \cdots & \boldsymbol{h}'_k[i-M+1] \end{pmatrix}^T$$

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

$$\boldsymbol{H}_k[i] := [\boldsymbol{\delta}_k[i, i-1]\ \boldsymbol{\delta}_k[i, i-2]\ \cdots\ \boldsymbol{\delta}_k[i, i-Q[i]]\ \mathbf{1}_M]$$

$$\boldsymbol{e}_k[i] = \boldsymbol{H}_k[i]^T \boldsymbol{f}_k[i] - [0\ \cdots\ 0\ 1]^T$$

$$\boldsymbol{\Delta}_k[i] = 2\boldsymbol{H}_k[i]\boldsymbol{e}_k[i]$$

$$\boldsymbol{f}_k[i+1] = P_{\mathcal{C}}(\boldsymbol{f}_k[i] - \alpha_k[i]\boldsymbol{\Delta}_k[i])$$

    end

  end

$$\begin{bmatrix} \boldsymbol{h}_1[i+1] \\ \vdots \\ \boldsymbol{h}_N[i+1] \end{bmatrix} = \boldsymbol{P}[i] \begin{bmatrix} \boldsymbol{h}'_1[i] \\ \vdots \\ \boldsymbol{h}'_N[i] \end{bmatrix}, \qquad \begin{bmatrix} \boldsymbol{h}^f_1[i+1] \\ \vdots \\ \boldsymbol{h}^f_N[i+1] \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1[i+1]^T \boldsymbol{y}_1[i+1] \\ \vdots \\ \boldsymbol{f}_N[i+1]^T \boldsymbol{y}_N[i+1] \end{bmatrix}$$

end

intuitive and based on NIP that takes the form

$$p_m(\lambda) = \sum_{l=0}^{m} \beta_l \lambda^l = \beta_0 + \beta_1 \lambda + \beta_2 \lambda^2 + \cdots + \beta_m \lambda^m \tag{25}$$

where m is the polynomial order and $\beta_l, l = 0, \cdots, m$ are the coefficients of the filter $\boldsymbol{f}_m$, defined as:

$$\boldsymbol{f}_m = \begin{pmatrix} \beta_0 & \beta_1 & \cdots & \beta_m \end{pmatrix}^T \tag{26}$$

DOI: 10.1002/dac

The approach assumes that the spectrum of $\boldsymbol{W}$ lies in an interval $[a, 1]$, and the coefficients of the filter can be evaluated using the following smoothness constraints at the left end point:

$$p_m(a) = 0, \quad p_m(1) = 1, \quad p_m^{(i)}(a) = 0, \; i = 1, \cdots, m-1 \tag{27}$$

The second approach in [9] is based on minimizing the spectral radius of the consensus matrix $\boldsymbol{W}$ defined as [24]:

$$\rho\left(p_m(\boldsymbol{W}) - (\mathbf{1}\mathbf{1}^T/n)\right) = \|p_m(\boldsymbol{W}) - (\mathbf{1}\mathbf{1}^T/n)\|_2 \tag{28}$$

using SDP solving the following optimization problem:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{m+1}} \rho &\left(\sum_{l=0}^{m} \beta_l \boldsymbol{W}^l - (\mathbf{1}\mathbf{1}^T/n)\right) \\ \text{subject to} &\left(\sum_{l=0}^{m} \beta_l \boldsymbol{W}^l\right)\mathbf{1} = \mathbf{1} \end{aligned} \tag{29}$$

The suggested algorithm for accelerating the consensus in the ASL problem using non-adaptive polynomial filters is given in Table IV.

Remark 3

On the contrary to the adaptive filters evaluated using algorithms II and III in Tables II and III, the non-adaptive polynomial filter evaluated in Table IV is applied on the node estimates *before* updating its values through the compact averaging formula using the extended consensus matrix.

7. SIMULATION STUDY

In this simulation study, we consider a sensor network with 5000 nodes randomly distributed in 100 m × 100 m field. A measure of the acoustic source energy is generated using the model in (9) with $A = 100$ and $\sigma_{v_k}^2 = 0.5$. The source is located at $\boldsymbol{h}^* = (50\ 50)^T$ and at each realization only nodes with $y_k \geqslant 5$ participate in the estimation task. These settings are similar to those in [17,18] for comparison sake. For brevity, the network is static, and the graph is undirected at each realization. The weight $w_{k,l}$ $(= w_{l,k})$ of the edge (k, l) is the Metropolis–Hastings weight [25]

$$w_{k,l} = \begin{cases} 1/\max(g_k, g_k) & \text{if } k \neq l \text{ and } l \in \mathcal{N}_k \\ 1 - \sum_{l \in \mathcal{N}_k \setminus \{k\}} w_{k,l} & \text{if } k = l \\ 0 & \text{otherwise} \end{cases}$$

where $g_k = |\mathcal{N}_k|$ is the degree for node k. Note that in this case, the extended matrix $\boldsymbol{P}$ is given by $\boldsymbol{P} = \boldsymbol{W} \otimes \boldsymbol{I}_2 \in \mathbb{R}^{2N \times 2N}$, where the component of the kth row and lth column of $\boldsymbol{W}$ is given by $w_{k,l}$.

The goal of the suggested algorithms in this paper is to estimate the location of the acoustic source $\boldsymbol{h}^*$. Therefore, to compare different algorithms, the average mean square error (MSE) and the average mean square distance to consensus (MSDC) are used as a measure of how far the network is from achieving consensus. The average MSE and average MSDC are defined as:

$$\begin{aligned} \text{MSE}[i] &= \mathrm{E}\left[(1/N)\|\boldsymbol{\psi}[i] - \mathbf{1}_N \otimes \boldsymbol{h}^*\|^2\right] \\ \text{MSDC}[i] &= \mathrm{E}\left[(1/N)\|(\boldsymbol{I}_{2N} - \boldsymbol{B}\boldsymbol{B}^T)\boldsymbol{\psi}[i]\|^2\right] \end{aligned} \tag{30}$$

where E denotes the expectation, $\boldsymbol{\psi}[i] = \left(\boldsymbol{h}_1[i]^T\ \boldsymbol{h}_2[i]^T\ \cdots\ \boldsymbol{h}_N[i]^T\right)^T$ and $\boldsymbol{B}$ is defined in Lemma 1 (*cf.* Section 3). In the following, ensemble average curves are obtained from the average of 100 realizations.

Table IV. Algorithm IV: acoustic source localization using adaptive projected subgradient method and Newton interpolation polynomial or semi-definite programming.

Choose polynomial order m for:

$$p_m(\lambda) = \sum_{l=0}^{m} \beta_l \lambda^l = \beta_0 + \beta_1\lambda + \beta_2\lambda^2 + \cdots + \beta_m\lambda^m$$

Evaluate the polynomial coefficients using:

$$p_m(a) = 0, \quad p_m(1) = 1, \quad p_m^{(i)}(a) = 0, \; i = 1, \cdots, m-1$$

or solving $\quad \min_{\beta\in\mathbb{R}^{m+1}} \rho\left(\sum_{l=0}^{m} \beta_l \boldsymbol{W}^l - (\mathbf{1}\mathbf{1}^T/n)\right)$

$\quad$ subject to $\left(\sum_{l=0}^{m} \beta_l \boldsymbol{W}^l\right)\mathbf{1} = \mathbf{1}$

Construct the non-adaptive filter $\boldsymbol{f}_m = (\beta_0 \; \beta_1 \; \cdots \; \beta_m)^T$

for time $i = 0, 1, \cdots$

for $k = 1 : N$. Choose $\delta_k > 0$, $\varepsilon = \sqrt{A/(y_k - e_k[i])}$ and $\mu_k[i] \in (0, \mathcal{M}_k[i])$ where

$$\mathcal{M}_k[i] = \begin{cases} 1 & \text{if } \boldsymbol{h}_k[i] \in \bigcap_{l\in\mathcal{N}_k} C_l[i] \\ \frac{\sum_{l\in\mathcal{N}_k} \|\kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2}{\|\sum_{l\in\mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\|^2 + \delta_k[i]} & \text{otherwise} \end{cases}$$

$$P_{C_l[i]}(\boldsymbol{h}) = \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{h} \in C_l[i] \\ \boldsymbol{r}_k + \frac{\varepsilon}{\|\boldsymbol{h} - \boldsymbol{r}_k\|}(\boldsymbol{h} - \boldsymbol{r}_k) & \text{otherwise} \end{cases}$$

$$\boldsymbol{h}'_k[i] = \boldsymbol{h}_k[i] + \mu_k[i] \; \left(\sum_{l\in\mathcal{N}_k} \kappa_{l,k} P_{C_l[i]}(\boldsymbol{h}_k[i]) - \boldsymbol{h}_k[i]\right)$$

if $\mod(i, m+1) == 0$

$$\boldsymbol{y}_k[i] := \begin{pmatrix} \boldsymbol{h}'_k[i] & \boldsymbol{h}'_k[i-1] & \cdots & \boldsymbol{h}'_k[i-M+1] \end{pmatrix}^T, \quad \tilde{\boldsymbol{h}}_k[i] = \boldsymbol{f}_m^T \boldsymbol{y}_k[i]$$

else $\quad \tilde{\boldsymbol{h}}_k[i] = \boldsymbol{h}'_k[i]$

end if

end

$$\begin{bmatrix} \boldsymbol{h}_1[i+1] \\ \vdots \\ \boldsymbol{h}_N[i+1] \end{bmatrix} = \boldsymbol{P}[i] \begin{bmatrix} \tilde{\boldsymbol{h}}_1[i] \\ \vdots \\ \tilde{\boldsymbol{h}}_N[i] \end{bmatrix}$$

end

In these simulations, the performance of the two versions of the APSM (algorithm I), namely the APSM using disks with fixed radius, $e_k[i] = 0$ and $\mu_k[i] = 0.2$ (denoted as algorithm I-A), and the APSM using disks with time-varying radius, $e_k[i] = i/10$ and $\mu_k[i] = 0.99$ (denoted as algorithm I- B) are compared with the performance when the filter is used. These chosen design values are

DOI: 10.1002/dac

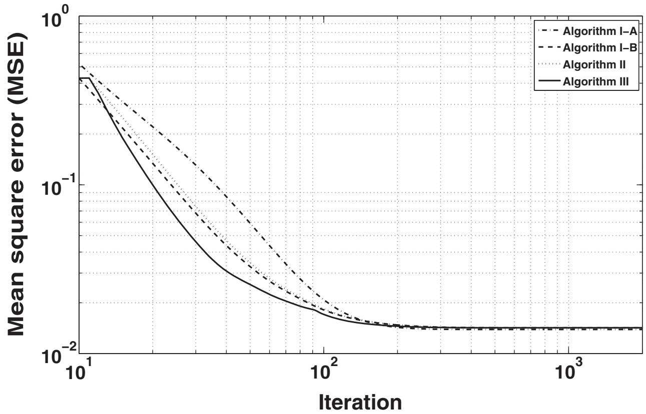

Figure 1. Mean square error (MSE) for Algorithms I, II, and III.

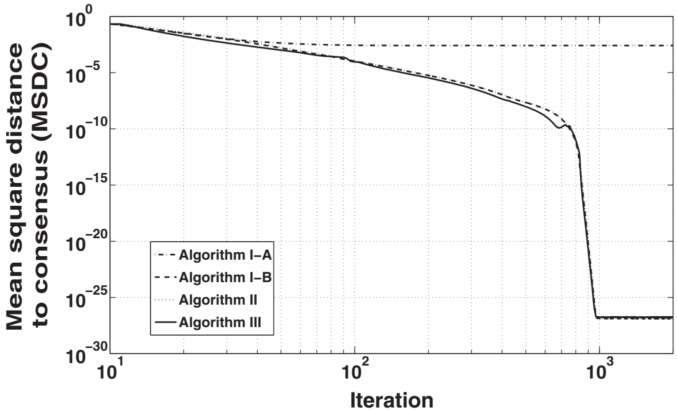

Figure 2. Mean square distance to consensus (MSDC) for Algorithms I, II, and III.

similar to the ones used in [17] that achieve much better performance than the incremental projection onto convex sets algorithm [18] with a relaxation parameter chosen as one in the first 30 iterations and then it decreases to $1/i$. As concluded in [17], algorithm I-B gives the best performance for the ASL problem.

The simulation results using the APSM and CAPA (algorithm II) and using the APSM and SGA (algorithm III) are shown in Figures 1–2. The filter length was chosen as $M = 9$, and the number of past vectors $\boldsymbol{y}_k[i]$ that should be stored in the node memory was 7 ($J = 6$). The step sizes were $\gamma_k[i] = 0.01$, and $\alpha_k[i] = 0.005$. These results show that algorithm III achieves better performance than algorithm II but with similar estimation accuracy as algorithm I. This is expected because all the approaches are based on the principle of projection over convex sets. As it is noticed from Figure 1, algorithm III does not improve or accelerate the consensus. This is believed to be due to the numerical problems encountered while evaluating the pseudoinverse in (21). To avoid instability of the CAPA, the small singular values of the matrix $\boldsymbol{H}_k[i]$ are treated as zeros [21]. This action will add some approximation errors that prevent the algorithm from achieving more accurate estimates in the transient phase. These errors were avoided in algorithm III where SGA is used.

On the other hand, the simulation results of the two versions of algorithm IV, namely using the APSM and Newton's interpolating polynomial (denoted as algorithm IV-NIP) and using the APSM and SDP (denoted as algorithm IV-SDP) are given in Figures 3–4. Also, here the filter length was chosen to be 9 ($m = 8$). For algorithm IV-NIP, the MATLAB Symbolic Toolbox was used to evaluate the polynomial coefficients in (25) using the smoothness constraints in (27). For algorithm IV-SDP, the optimization problem in (29) was solved using CVX [26]. As shown in Figures 3–4, the best

DOI: 10.1002/dac

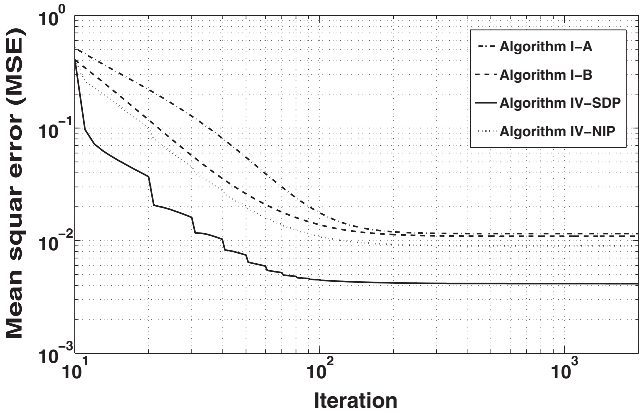

Figure 3. Mean square error (MSE) for Algorithms I and IV.

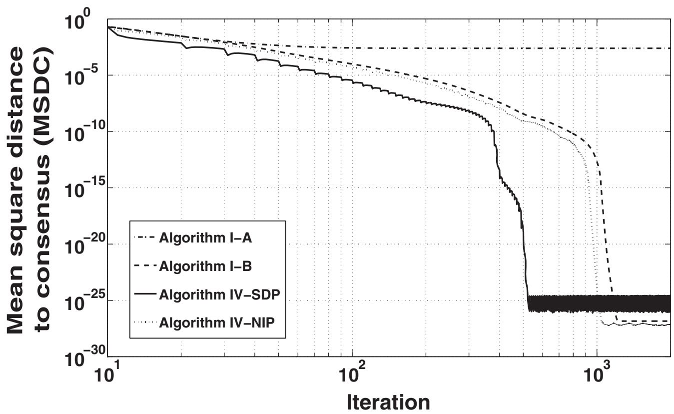

Figure 4. Mean square distance to consensus (MSDC) for Algorithms I and IV.

performance is achieved using algorithm IV-SDP because it makes use of the assumption that the network topology is known beforehand, but in algorithm IV-NI, only the spectrum of the consensus matrix is assumed to lie in the interval $[a, 1]$.

8. CONCLUSIONS

Accelerating consensus based on adaptive and non-adaptive filtering techniques have been considered in this paper. The basic idea is to store a number (chosen by the user) of consecutive samples of the local information at each sensor and design a filter to be applied on these samples to accelerate consensus through the diffusion network. This filter can be designed in adaptive way using CAPA or SGA. The SGA is more simpler and more attractive from performance, computational burden, and numerical arithmetic point of views. Also, the filter can be designed in a non-adaptive way using NIPs or SDP provided that the consensus matrix (network topology) is known beforehand. The comparative simulation studies in this paper show that the best performance for accelerating the consensus in the diffusion network based on the adaptive subgradient algorithm can be achieved using non-adaptive filter and SDP (algorithm IV-SDP). An interesting point for future research is to relax the assumption of knowing the network topology beforehand to make the algorithm fully distributed.

ACKNOWLEDGEMENTS

This work was supported by UK-MOD University Defence Research Center (UDRC) in signal processing, Imperial College, London, UK. The authors would also like to thank Dr. R. Cavalcante for valuable discussions on this work.

REFERENCES

1. Akyildiz I, Su W, Sankarasubramaniam Y, Cayirci E. Survey on sensor networks. *IEEE Signal Processing Magazin* 2002; **40**(8):102–114.
2. Jang KW. Meta-heuristic algorithms for channel scheduling problem in wireless sensor networks. *International Journal of Communication Systems* 2012; **25**(4):427–446.
3. Tang J, Dai S, Li J, Li S. Gossip-based scalable directed diffusion for wireless sensor networks. *International Journal of Communication Systems* 2011; **24**(11):1418–1430.
4. Khan AUR, Madani SA, Hayat K, Khan SU. Clustering-based power-controlled routing for mobile wireless sensor networks. *International Journal of Communication Systems* 2012; **25**(4):529–542.
5. Tubaishat M, Madria S. Sensor networks: an overview. *IEEE Potentials* 2003; **22**(2):20–23.
6. Sung TW, Yang CS. An adaptive joining mechanism for improving the connection ratio of zigbee wireless sensor networks. *International Journal of Communication Systems* 2010; **23**(2):231–251.
7. Xiong NX, Cao M, Vasilakos AV, Yang LT, Yang F. An energy-efficient scheme in next-generation sensor networks. *International Journal of Communication Systems* 2010; **23**(9,10):1189–1200.
8. Kokiopoulou E, Frossard P. Accelerating distributed consensus using extrapolation. *IEEE Signal Processing Letters* 2007; **14**(10):665–668.
9. Kokiopoulou E, Frossard P. Polynomial filtering for fast convergence in distributed consensus. *IEEE Transactions on Signal Processing* 2009; **57**(1):342–354.
10. Cavalcante RLG, Mulgrew B. Adaptive filter algorithms for accelerated discrete-time consensus. *IEEE Transactions on Signal Processing* 2010; **58**(3):1049–1058.
11. Golub GH, Loan CFV. *Matrix Computations*, 3rd edn., The John Hopkins Studies in Mathematical Sciences: Baltimore, MD , USA, 1996.
12. Stark H, Yang Y. *Vector Space Projections - A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*. Wiley: New York, USA, 1998.
13. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press: Cambridge, UK, 2006.
14. Bertsekas DP, Nedic A, Ozdaglar AE. *Convex Analysis and Optimization*. Athena Scientific Press: USA, 2003.
15. Cavalcante RLG, Yamada I, Mulgrew B. An adaptive projected subgradient approach to learning in diffusion networks. *IEEE Transactions on Signal Processing* 2009; **57**(7):2762–2774.
16. Yamada I, Ogura N. Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions. *Numerical Functional Analysis and Optimization* 2004; **25**(7/8):593–617.
17. Cavalcante RLG, Yamada I, Mulgre B. Learning in diffusion networks with an adaptive projected subgradient method. *Proceedings of the 34th ICASSP*, Taipei, Taiwan, April 19-24, 2009; 2853–2856.
18. Blatt D, Hero AO, III. Energy-based sensor network source localization via projection onto convex sets. *IEEE Transactions on Signal Processing* 2006; **54**(9):3614–3619.
19. Slavakis K, Yamada I, Ogura N, Yukawa M. Adaptive projected subgradient method and set theoretic adaptive filtering with multiple convex constraints. *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, USA, November 2004; 960–964.
20. Theodoridis S, Slavakis K, Yamada I. Adaptive learning in a world of projections. *IEEE Signal Processing Magazine* 2011; **28**(1):97–123.
21. Golub GH, Loan CFV. *Matrix Computations*. Cambridge University: Cambridge, UK, 1996.
22. Söderström T, Stoica P. *System Identification*. Prentice-Hall International: Hemel Hempstead, United Kingdom, 1989.
23. Ljung L. *System Identification - Theory for the User*, Second edn., Prentice-Hall: Upper Saddle River, NJ , USA, 1999.
24. Boyd S, Xiao L. Fast linear iterations for distributed averaging. *Systems Control Letters* 2004; **53**:65–78.
25. Lopes CG, Sayed AH. Diffusion least-mean squares over adaptive networks: formulation and performance analysis. *IEEE Transactions on Signal Processing* 2008; **56**(7):3122–3136.
26. Grand M, Boyd S. CVX: Matlab software for disciplined convex programming (web page and software), 2008. (Available from: http://stanford.edu/boyd/cvx).

DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

Emad Abd-Elrady was born in Cairo, Egypt, in 1970. He received the B.Sc. and M.Sc. degrees in Electrical Engineering from Ain Shams University, Cairo, Egypt in 1993 and 1997, respectively. He received the Licentiate and Ph.D. degrees in Electrical Engineering with specialization in Signal Processing from Division of Systems and Control, Uppsala University, Sweden in 2002 and 2005, respectively. From January 2006 to March 2009, he was a senior researcher and project leader at Christian Doppler Laboratory for Nonlinear Signal Processing, Graz University of Technology, Austria where he was working in a cooperation project with Infineon Technologies, Villach. The project focuses on realization and implementation aspects of adaptive digital compensation methods for analog nonlinearities in VLSI circuits for broadband communications. From March 2010 to January 2011, he was a research fellow at Institute for Digital Communications, The University of Edinburgh, Scotland, UK, where he was working in a Ministry of Defence project on distributed sensor network. Since February 2011, he has been electronics engineering faculty at Abu Dhabi Women's College, United Arab Emirates. His research interests include adaptive filtering, system identification, adaptive, nonlinear and distributed signal processing, dynamical system modeling, distributed and convex optimization, and wireless sensor networks.

Bernard Mulgrew received his BSc degree in 1979 from Queen's University Belfast. After graduation, he worked for 4 years as a development engineer in the Radar Systems Department at Ferranti, Edinburgh. From 1983 to1986, 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. He received his PhD and was appointed to a lectureship in 1987. He currently holds the Selex ES/Royal Academy of Engineering Research Chair in Signal Processing. His research interests are in adaptive signal processing and estimation theory and in their application to radar and distributed sensing systems. He is a co-author of three books on signal processing and over 80 journal papers. He is a Fellow of the IEEE, a Fellow of the Royal Academy of Engineering, a Fellow of the Royal Society of Edinburgh, and a Fellow of the IET.

DOI: 10.1002/dac