

Accelerated Distributed Average Consensus Via Localized Node State Prediction

Tuncer C. Aysal, Boris N. Oreshkin and Mark J. Coates

Telecommunications and Signal Processing–Computer Networks Laboratory

Department of Electrical and Computer Engineering

McGill University, Montreal, QC, Canada

Email: {tuncer.aysal, mark.coates}@mcgill.ca, boris.oreshkin@mail.mcgill.ca

Abstract

The problem of distributed consensus has recently received a lot of attention, particularly in the framework of ad hoc sensor networks. The average consensus problem in the distributed signal processing context is addressed by linear iterative algorithms, with asymptotic convergence to the consensus. The convergence of the average consensus for an arbitrary weight matrix satisfying the convergence conditions is unfortunately slow refraining the use of the developed algorithms in applications. In this paper, we propose the use of extrapolation methods in order to accelerate distributed linear iterations. We utilize a linear operator to predict the future node state values and then combine the prediction with the current node state value in a convex fashion driving overall system state closer to the true consensus value faster than the standard consensus algorithms. A faster convergence is, hence, achieved by the bypassing of redundant states. The proposed method is linear and computationally effective. We focus on a special case of the proposed framework and derive the optimal mixing parameter. Noting that the optimal mixing parameter requires knowledge about the eigenvalues of the arbitrary weight matrix, we present a bound on the optimal parameter requiring only local information, and prove the validity of the suboptimal solution in the practical cases by showing that its performance is close-to-optimal and it is feasible in practical scenarios. Finally, we provide simulation results that demonstrate the validity and effectiveness of the proposed scheme. These results also indicate that in general situation the consensus based on the proposed approach significantly outperforms the optimum algorithm based on weight matrix optimization relying on semidefinite programming paradigm.

Index Terms

distributed signal processing, average consensus, linear prediction.

I. INTRODUCTION

In both wireless sensor and peer-to-peer networks, there is interest in simple protocols for computing aggregate statistics [1]–[4]. Distributed average consensus, hence, is an important issue in sensor networks. There are several simple methods for distributed average consensus. In this paper we focus on a particular class of iterative algorithms for average consensus: each node updates its state by adding a weighted sum of the local nodes and these weights are algorithm parameters [5]–[7]. The state at each node in the iteration consists of a single real number, which overwrites the previous value. The algorithm is time-independent, *i.e.*, does not depend on t . The algorithm computes the average asymptotically [8].

Ad hoc networks of autonomous sensors and actuators are attractive solutions for a broad range of applications. Such networks find use in civilian and military applications, including target tracking and surveillance for robot navigation, source localization, weather forecasting, medical monitoring and imaging. Distributed average consensus, in ad hoc networks, is an important issue in distributed agreement and synchronization problems [9] and is also a central topic for load balancing (with divisible tasks) in parallel computers [10], [11]. More recently, it has also found applications in distributed coordination of mobile autonomous agents [12], [13] and distributed data fusion in sensor networks [5], [6], [14]. In general, the networks envisioned for many of these applications involve large numbers of possibly randomly distributed inexpensive sensors. A major drawback of the developed average consensus algorithms is the number of iterations taking to converge to consensus often refraining the use of them in practical scenarios.

A. Related Work

Much of the work dealing with consensus algorithm acceleration has been done by the authors of [5], [8], [15]. They showed that it is possible to formulate the problem of asymptotic convergence time minimization as a convex semidefinite one. The solution to this problem is then obtained in the form of optimum weight matrix resulting from a matrix optimization algorithm. The disadvantages of this approach are twofold. Firstly, the approach is based on convex optimization paradigm and the time or computational resources necessary to set up the network may be quite substantial. Secondly, this approach requires connectivity pattern to be known in advance and thus assumes that there is a fusion center or some distributed mechanism that is aware of the global network state. To combat the second problem the use of iterative optimization utilizing subgradient algorithm is proposed in [15]. However, calculation of subgradient still requires the knowledge of the eigenvector corresponding to the second largest eigenvalue of the weight matrix. To make the algorithm distributed authors of [15] use decentralized orthogonal iterations [16] for eigenvector calculation. The resulting algorithm is extremely demanding in terms of time, computation, and communication, because as a matter of fact it consists of two consensus procedures. Another approach to weight

matrix optimization is to set the neighbouring edge weights to be equal to some constant. The optimization of this constant with respect to minimizing the asymptotic convergence time gives the value of the constant inversely proportional to the sum of the largest and the second smallest eigenvalues of the Laplacian spectrum. This again implies that connectivity pattern is known to some weight matrix construction mechanism. The weight matrix constructed using this mechanism is called the best constant weight matrix [8]. The suboptimality of the best constant weight matrix stems from the fact that all the edge weights are constrained to be the same.

B. Summary of Contributions

In this paper, we propose accelerating the convergence rate of a distributed average consensus operating with an arbitrary weight matrix satisfying the required convergence conditions [5], [7] by using a convex combination of the values obtained by a linear predictor and consensus iteration. Unlike the previous methods, we hence do not burden the nodes with extra computational load since the prediction is linear and its parameters can be calculated offline. We present a general framework of accelerating the consensus and focusing on a special case to gain further insight, we derive the optimal convex combination parameter when approached from the asymptotic convergence rate perspective. Noting that the optimal parameter requires the knowledge of the second largest and the smallest eigenvalues of the weight matrix, we derive suboptimal solutions demanding much less information and easily implementable in practical scenarios. We prove the validity of the proposed suboptimal approach by showing that it is feasible to obtain the suboptimal parameter for the weight matrix considered in practice and its close-to-optimal performance. Finally, we report simulation results evaluating the behavior and characteristics of the proposed optimal and suboptimal approaches in varying scenarios. These results show that the generalized version of our algorithm outperforms the optimum consensus algorithm based on weight matrix optimization when grid search for unknown optimum value of mixing parameter is used.

C. Paper Organization

The remaining of this paper is organized as follows. Section II introduces the distributed average consensus problem and formulates the proposed framework to improve the rate of convergence. The proposed algorithm, along with its properties and the optimal mixing parameter for a simplified case, the provided rate of convergence improvement, a practical suboptimal solutions along with their applicability to realistic sensor networks are detailed in Section III. We report the numerical examples testing the proposed algorithms in Section IV. Finally, section V concludes the paper.

II. PROBLEM FORMULATION

This section first introduces elementary graph theory followed by a brief review of the standard consensus algorithm and formulates the proposed framework to the consensus acceleration. Considered next is the proposed approach to accelerate the consensus algorithm for any graph.

We define a graph $G = (\mathcal{V}, \mathcal{E})$ as 2-tuple, consisting of a set \mathcal{V} with $|\mathcal{V}| = N$ vertices, where $|\cdot|$ denotes the cardinality, and a set \mathcal{E} with $|\mathcal{E}| = M$ edges. We denote an edge between vertices i and j as an unordered pair $(i, j) \in \mathcal{E}$. The presence of an edge between two vertices indicates that they can establish bidirectional noise-free communication with each other. We assume that transmissions are always successful and that the topology is fixed. We assume connected network topologies and the connectivity pattern of the graph is given by the $N \times N$ adjacency matrix $\Phi = [\Phi_{ij}]$, where

$$\Phi_{ij} = \begin{cases} 1 & \text{if } (i, j) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases}. \quad (1)$$

Moreover, we denote the neighborhood of the node i by, $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. Also, the degree of the node i is given by $d_i \triangleq |\mathcal{N}_i|$.

We consider a set of nodes of a network (vertices of the graph), each with an initial real valued scalar $x_i(0)$, where $i = 1, 2, \dots, N$. Let $\mathbf{1}$ denote the vector of ones. The goal is to develop a distributed iterative algorithm that computes at every node in the network, the value $\bar{\mathbf{x}} \triangleq (N)^{-1} \mathbf{1}^T \mathbf{x}(0)$. In this paper we focus on a particular class of iterative algorithms for average consensus, widely used in the applications cited above. Each node updates its state by adding a weighted sum of the local nodes, *i.e.*,

$$x_i(t+1) = W_{ii}x_i(t) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t) \quad (2)$$

for $i = 1, 2, \dots, N$ and $t = 0, 1, \dots$. Here W_{ij} is a weight associated with the edge $\{i, j\}$ and N is the total number of nodes. These weights are algorithm parameters [5], [6]. At $t = 0$ (after all sensors have taken the measurement), each node initializes its state as $x_i(0)$. At each following step, each node updates its state with a linear combination of its own state and the states at its neighbors. Moreover, setting $W_{ij} = 0$ whenever $\Phi_{ij} = 0$, the distributed iterative process reduces to the following recursion

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t) \quad (3)$$

where $\mathbf{x}(t)$ denotes the state vector. The weight matrix, \mathbf{W} , needs to satisfy the following conditions to ensure asymptotic average consensus [7]:

$$\mathbf{W}\mathbf{1} = \mathbf{1}, \mathbf{1}^T\mathbf{W} = \mathbf{1}^T, \rho(\mathbf{W} - \mathbf{J}) < 1 \quad (4)$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix:

$$\rho(\mathbf{W}) \triangleq \max_i \{|\lambda_i| : i = 1, 2, \dots, N\}. \quad (5)$$

Here $\{\lambda_i\}_{i=1}^N$ denote the eigenvalues of \mathbf{W} . In this paper, we assume that, in the modulus, the second largest eigenvalue of the weight matrix is $\lambda_{(2)}$, *i.e.*, $\lambda_{(2)} > |\lambda_{(N)}|$, where $\lambda_{(i)}$ denotes the i -th ranked eigenvalue. We make this assumption to simplify the presentation, but the results can be easily extended to avoid this simplification. Note that the weight matrices satisfying the required convergence conditions are proposed if the underlying graph is connected and non-bipartite, *e.g.*, Maximum-degree and Metropolis weights [5], [7]. These schemes also satisfy the fact that the second largest eigenvalue, in the modulus, is $\lambda_{(2)}$ [5]–[7], [15].

We modify the above consensus algorithm to increase its convergence speed in the following way. State value vector $\mathbf{x}(t)$ is a possibly time-variant function g_t of the values $\mathbf{x}^W(t)$ and $\mathbf{x}^P(t)$:

$$\mathbf{x}(t) = g_t(\mathbf{x}^W(t), \mathbf{x}^P(t; k)) \quad (6)$$

where $\mathbf{x}^W(t)$ is the local node state propagation similar to the standard consensus procedure:

$$\mathbf{x}^W(t) = h_t(\mathbf{x}(t-1)) \quad (7)$$

where h_t is some possibly time-variant and non-linear function. Moreover, $\mathbf{x}^P(t; k)$ is a k -step prediction of future node states obtained from M previous local node states $\mathbf{x}_i(t) = [x_i(t-M+1), \dots, x_i(t-1), x_i^W(t)]^T$:

$$\mathbf{x}^P(t; k) = f_t(\{\mathbf{x}_i(t) : i = 1, 2, \dots, N\}) \quad (8)$$

where f_t is some predictor.

The rationale behind the proposed system configuration outlined by equations (6–8) is as follows: we note, by observing the individual node trajectories produced by the standard consensus algorithm [8], that the convergence curves are smooth. Thus, it is reasonable to expect that it is possible to predict future local node state from a collection of previous node states. If the prediction is accurate enough then it could be combined with the current node state to drive overall system state closer to the true consensus value faster than the standard consensus algorithms. A faster convergence could, hence, be achieved by virtually bypassing the redundant states.

III. ACCELERATING DISTRIBUTED AVERAGE CONSENSUS FOR AN ARBITRARY WEIGHT MATRIX

In the following, we detail the general proposed predictor based distributed average consensus and to gain further insight about the proposed algorithm focus on a special case yielding computationally simple, linear and closed-form attractive solution. Moreover for the considered special case, we present an optimization technique for the

mixing parameter when approached from a rate of convergence perspective, *i.e.*, a closed-form solution maximizing the rate of convergence.

A. Predictor Based Distributed Average Consensus

Computational resources available at the nodes are often scarce and it is desirable that the algorithms designed for distributed signal processing are computationally inexpensive. Linear functions are comparatively easy to implement. Moreover, the analysis of linear algorithms is often straightforward and expressions for performance metrics and iteration outcomes can be derived in closed form. This often makes algorithms based on linear approximations more predictable, computationally efficient and attractive. Thus in the following we will consider linear matrix functions of appropriate size $h_t \triangleq \mathbf{W}$, $g_t \triangleq [1 - \alpha, \alpha]^T$, $f_t \triangleq \Theta$. With these in mind, equations (6–8), when node specific versions are considered, take the following form:

$$x_i(t) = \alpha x_i^P(t) + (1 - \alpha)x_i^W(t) \quad (9a)$$

$$x_i^W(t) = W_{ii}x_i(t-1) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t-1) \quad (9b)$$

$$x_i^P(t) = \Theta_{\mathbf{x}_i(t)}^T \mathbf{x}_i(t) \quad (9c)$$

where $\Theta_{\mathbf{x}_i(t)} = [\theta_{\mathbf{x}_i(t)}, \dots, \theta_{\mathbf{x}_i(t)}]^T$ is curve parameter vector. We consider a convex combination of the predicted and obtained states, *i.e.*, $\alpha \in [0, 1]$. Convergence curve can be linearized in the vicinity of current state

$$x_i(t) = \theta_{\mathbf{x}_i(t)} t + \theta_{\mathbf{x}_i(t)}. \quad (10)$$

Parameters $\theta_{\mathbf{x}_i(t)}$ and $\theta_{\mathbf{x}_i(t)}$ of this linear approximation can be estimated from the collection of previous node states $\mathbf{x}_i(t)$ using standard least squares procedure [17]:

$$\mathbf{x}_i(t) = \mathbf{A}_t \Theta_{\mathbf{x}_i(t)} \quad (11)$$

where \mathbf{A}_t is a $M \times 2$ matrix

$$\mathbf{A}_t = \begin{bmatrix} t - M + 1 & 1 \\ \dots & \dots \\ t - 1 & 1 \\ t & 1 \end{bmatrix} \quad (12)$$

Solution to the matrix equation (11) is known to have the following form:

$$\Theta_{\mathbf{x}_i(t)} = (\mathbf{A}_t^T \mathbf{A}_t)^{-1} \mathbf{A}_t^T \mathbf{x}_i(t) = \mathbf{A}_t^\dagger \mathbf{x}_i(t) \quad (13)$$

Where \mathbf{A}_t^\dagger is the Moore-Penrose pseudoinverse of \mathbf{A}_t . Thus k -step prediction of local node state can be regarded as a simple linear extrapolation:

$$x_i^P(t, k) = \Theta_{\mathbf{x}_i(t)}^T \mathbf{t}_t^{P,k} \quad (14)$$

with $\mathbf{t}_t^{P,k}$ denoting the predictive time vector at time t :

$$\mathbf{t}_t^{P,k} = \begin{bmatrix} t + k \\ 1 \end{bmatrix} \quad (15)$$

It should be noted that an equivalent extrapolation procedure can be formulated in terms of matrix \mathbf{A} and predictive time vector $\mathbf{t}^{P,k}$ having simplified time invariant form:

$$\mathbf{A} \triangleq \begin{bmatrix} 1 & 1 \\ \cdots & \cdots \\ M-1 & 1 \\ M & 1 \end{bmatrix} \quad \mathbf{t}^{P,k} \triangleq \begin{bmatrix} M+k \\ 1 \end{bmatrix} \quad (16)$$

Reorganizing equations (13,14) and taking into account (16) gives the following simple node state prediction procedure:

$$\Theta = \mathbf{A}^{\dagger T} \mathbf{t}^{P,k} \quad (17)$$

subsequently yielding

$$x_i^P(t) = \Theta^T \mathbf{x}_i(t). \quad (18)$$

Remark 1: In the following, we note two important remarks regarding the proposed approach for accelerating the standard consensus algorithm for any given weight matrix:

- (i) It can be seen from (17) that $x_i^P(t)$ is a linear combination of M previous local consensus values. Thus the consensus acceleration mechanism outlined in equations (9a–9c) is fully local if it is possible to find optimum value of α in (9a) that does not require any global knowledge.
- (ii) Note also that Θ can be calculated off-line as it does not depend on the data.

B. One Step Predictor Based Distributed Average Consensus

General expressions describing the algorithm (9a–9c) using multistep predictor operating on multiple previous node states are given in Appendix I. In this general case, the algorithm analysis is complicated, but to gain further insight to the algorithm's performance we analyze an important case when the algorithm (9) is based on one step extrapolator of node state operating on two previous node states, *i.e.*, $k = 1$ and $M = 2$. In this case $\Theta = [-1, 2]^T$

hence $x_i^P(t)$ can be expressed as follows:

$$x_i^P(t) = 2x_i^W(t) - x_i(t-1). \quad (19)$$

We note that the gradient of state can be estimated as $\widehat{\nabla}x_i(t) \triangleq x_i^W(t) - x_i(t-1)$. Thus (19) can be rewritten in a more concise form:

$$x_i^P(t) = x_i^W(t) + \widehat{\nabla}x_i(t). \quad (20)$$

It is of interest to note that the one-step predictor hence updates the current state in the gradient direction.

Substituting (19) into (9a) we get the following expression for $x_i(t)$:

$$x_i(t) = \alpha(2x_i^W(t) - x_i(t-1)) + (1-\alpha)x_i^W(t) \quad (21)$$

$$= (\alpha+1)x_i^W(t) - \alpha x_i(t-1) \quad (22)$$

$$= (\alpha+1) \left(W_{ii}x_i(t-1) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t-1) \right) - \alpha x_i(t-1) \quad (23)$$

$$= x_i(t-1)((1+\alpha)W_{ii} - \alpha) + (1+\alpha) \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t-1). \quad (24)$$

This can be written in matrix form as:

$$\mathbf{x}(t) = \mathbf{W}[\alpha]\mathbf{x}(t-1) \quad (25)$$

where $\mathbf{W}[\alpha]$ is the weight matrix (as a function of α) obtained through the proposed predictor based distributed average consensus algorithm:

$$\mathbf{W}[\alpha] \triangleq (1+\alpha)\mathbf{W} - \alpha\mathbf{I} \quad (26)$$

The following proposition describes some properties of the weight matrix $\mathbf{W}[\alpha]$. Specifically, we show that if the weight matrix \mathbf{W} satisfies the conditions necessary for asymptotical convergence, then $\mathbf{W}[\alpha]$ also guarantees the asymptotical convergence to consensus under some mild conditions.

Proposition 1: Suppose \mathbf{W} satisfies the necessary conditions for the convergence of the standard consensus algorithm. Moreover, let $\lambda_{(1)} \geq \lambda_{(2)} \geq \dots \geq \lambda_{(N)}$ denote the eigenvalues associated with eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$ and let $\lambda_{(i)}[\alpha]$ denote the ranked eigenvalues of $\mathbf{W}[\alpha]$.

- (i) If $\lambda_{(N)} \geq 0$, then $\mathbf{W}[\alpha]$ satisfies the required convergence conditions for all α . If $\lambda_{(N)} < 0$, then $\mathbf{W}[\alpha]$ is a doubly stochastic matrix. Moreover, $\rho(\mathbf{W} - \mathbf{J}) < 1$ if

$$\alpha < \frac{1 + \lambda_{(N)}}{1 - \lambda_{(N)}}. \quad (27)$$

- (ii) The predictor based weight matrix $\mathbf{W}[\alpha]$ has the same eigenvectors and its eigenvalues are related to the

eigenvalues of the original matrix \mathbf{W} via the relationship:

$$\lambda_{(i)}[\alpha] = (1 + \alpha)\lambda_{(i)} - \alpha \quad (28)$$

for any α and $i = 1, 2, \dots, N$.

Proof: See Appendix II. ■

As follows from the result of Proposition 1, eigenvalues of the predictor based weight matrix $\mathbf{W}[\alpha]$ experience left shift with respect to the eigenvalues of the original weight matrix \mathbf{W} when $\alpha > 0$. Moreover, it is easy to show that ordering of the eigenvalues does not change during the shift:

$$\lambda_{(i)} \leq \lambda_{(j)} \Rightarrow \lambda_{(i)}[\alpha] \leq \lambda_{(j)}[\alpha] \quad (29)$$

for all $i, j, \alpha \geq 0$, where i and j are associated with some eigenvectors $\mathbf{u}_i, \mathbf{u}_j$ of \mathbf{W} . The implications of Proposition 1 are as follows. First, the second largest and the smallest eigenvalues of matrix $\mathbf{W}[\alpha]$ always correspond to the second largest and the smallest eigenvalues of matrix \mathbf{W} . Second, the values of the second largest and the smallest eigenvalues of matrix $\mathbf{W}[\alpha]$ are always less than the values of the second largest and the smallest eigenvalues of matrix \mathbf{W} . Third, using the two previous facts and the definition of spectral radius (5) it is possible to formulate the problem of the mixing parameter optimization in a simple way. As follows from the considerations presented in the following subsection, this optimization problem is convex and has a simple closed form solution.

C. Optimization of The Mixing Parameter

Recall that α is the mixing parameter that defines the influence of the outcome of standard consensus iteration and predictor in (9a) on the value of current node state in the proposed accelerated consensus algorithm. In the following, we consider optimization of α when approached from the rate of convergence perspective. Optimum value of mixing parameter results in the maximum asymptotic convergence rate for algorithmic structure (9) and $M = 2, k = 1$ case.

Spectral radius $\rho(\mathbf{W}[\alpha] - \mathbf{J})$ defines the worst-case measure of asymptotic convergence rate [7]. In fact, spectral radius is directly related to the asymptotic convergence rate as defined in [8]:

$$r(\mathbf{W}) \triangleq \rho(\mathbf{W}[\alpha] - \mathbf{J}) = \sup_{\mathbf{x}(0) \neq \mathbf{J}\mathbf{x}(0)} \lim_{t \rightarrow \infty} \left(\frac{\|\mathbf{x}(t) - \mathbf{J}\mathbf{x}(0)\|}{\|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\|} \right)^{1/t}. \quad (30)$$

Thus the minimization of the spectral radius $\rho(\mathbf{W}[\alpha] - \mathbf{J})$ leads to the maximization of the convergence rate, or equivalently, to the minimization of the asymptotic convergence time $\tau \triangleq (\log(1/\rho(\mathbf{W}[\alpha] - \mathbf{J})))^{-1}$. The following theorem considers the optimum α when the goal is to maximize the asymptotic convergence rate.

Theorem 1: One step predictor based on two previous values of local node state has the fastest asymptotic

worst-case convergence rate if the value of mixing parameter equals to the following optimum value:

$$\alpha^* = \frac{\lambda_{(N)} + \lambda_{(2)}}{2 - \lambda_{(N)} - \lambda_{(2)}} \quad (31)$$

where $\lambda_{(i)}$ denotes the eigenvalues of the weight matrix \mathbf{W} .

Proof: See Appendix III. ■

Note that, as expected, optimal mixing parameter α^* satisfies the following:

$$\alpha^* < \frac{1 + \lambda_{(N)}}{1 - \lambda_{(2)} + 1 - \lambda_{(N)}} \quad (32)$$

$$< \frac{1 + \lambda_{(N)}}{1 - \lambda_{(N)}} \quad (33)$$

where both the first and second lines follow from the fact that $0 \leq \lambda_{(2)} < 1$, respectively. We can thus conclude that the optimal mixing parameter satisfies the required convergence conditions for all cases.

Remark 2: After some algebraic manipulations, it is of interest to note that

$$|\lambda_{(N)}[\alpha^*]| = \frac{\lambda_{(2)} - \lambda_{(N)}}{2 - \lambda_{(2)} - \lambda_{(N)}} = \lambda_{(2)}[\alpha^*]. \quad (34)$$

Note that the optimal mixing parameter provides just enough shift to the eigenvalues so that as we try to drive $\lambda_{(2)}[\alpha]$ smaller to obtain a faster convergence rate, optimal mixing parameter stops at a specific point where $|\lambda_{(N)}[\alpha^*]| = \lambda_{(2)}[\alpha^*]$ to avoid the $|\lambda_{(N)}[\alpha]| > \lambda_{(2)}[\alpha]$ case which would give a slower convergence rate.

D. Convergence Rate Analysis

To see to what extent the proposed algorithm (9) yields performance improvement over the conventional consensus, we consider the ratio of the spectral radius of corresponding matrices that gives the lower bound on performance improvement:

$$\gamma[\alpha] \triangleq \frac{\rho(\mathbf{W} - \mathbf{J})}{\rho(\mathbf{W}[\alpha] - \mathbf{J})} = \frac{\lambda_{(2)}}{\max\{\lambda_{(2)}[\alpha], |\lambda_{(N)}[\alpha]|\}} \quad (35)$$

The following proposition considers the provided convergence rate improvement over the standard consensus algorithm when optimal mixing parameter is utilized.

Proposition 2: In the optimal case, i.e., when $\alpha = \alpha^*$, performance improvement factor is given by

$$\gamma[\alpha^*] = \frac{\lambda_{(2)}(2 - \lambda_{(2)} - \lambda_{(N)})}{\lambda_{(2)} - \lambda_{(N)}}. \quad (36)$$

Proof: In the optimal case, substituting α^* into (35) and taking into account the fact that $|\lambda_{(N)}[\alpha^*]| = \lambda_{(2)}[\alpha^*]$, after some algebraic manipulations, yield the expression for $\gamma[\alpha^*]$. ■

Although (31) provides an expression for optimum mixing factor resulting in fastest asymptotic convergence rate, the calculation of this optimum value requires knowledge of the second and the last eigenvalues of matrix \mathbf{W} .

This in turn either requires knowledge of \mathbf{W} or some centralized mechanism for calculation and distribution of the eigenvalues of \mathbf{W} . In many practical situations such information may not be available. Therefore it is of interest to derive some suboptimum expressions for α that would result in less performance gain, but require considerably less information at the node level.

Proposition 3: The predictor based distributed average consensus has asymptotic worst-case convergence rate faster than that of conventional consensus if the value of mixing parameter is in the following range:

$$0 < \alpha \leq \alpha^*. \quad (37)$$

Proof: The asymptotic worst-case convergence rate of algorithm (9) is faster than that of conventional consensus algorithm if and only if $\gamma[\alpha] > 1 \Rightarrow \rho(\mathbf{W}[\alpha] - \mathbf{J}) < \rho(\mathbf{W} - \mathbf{J})$. We can rewrite this condition in the following form:

$$\begin{cases} \frac{\lambda_{(2)}(1 + \alpha) - \alpha}{\lambda_{(2)}} < 1 \\ \frac{\alpha(1 - \lambda_{(N)}) - \lambda_{(N)}}{\lambda_{(2)}} < 1 \end{cases} \quad (38)$$

indicating that

$$\begin{cases} \alpha(\lambda_{(2)} - 1) < 0 \\ \alpha < \frac{\lambda_{(2)} + \lambda_{(N)}}{1 - \lambda_{(N)}} \end{cases} \quad (39)$$

Taking into account that $(\lambda_{(2)} - 1) < 0$ the strengthened version of (39) can be written as follows:

$$0 < \alpha < \frac{\lambda_{(N)} + \lambda_{(2)}}{2 - \lambda_{(N)} - \lambda_{(2)}} \quad (40)$$

Finally, noting that this expression is equal to α^* concludes the proof. ■

To ensure that the performance improvement factor is greater than unity and remove the need of weight matrix information, we proceed to derive bounds on α^* satisfying the range defined by the Proposition 3. The mentioned constraints indicate that α^* needs to be lowerbounded, which, subsequently indicates that we need to derive a lowerbound for $\lambda_{(2)} + \lambda_{(N)}$. The next proposition provides such a bound for this term in terms of the trace of weight matrix \mathbf{W} .

Proposition 4: If the weight matrix \mathbf{W} satisfies the convergence conditions and its eigenspectrum is a convex function of the eigenvalue index then

$$\lambda_{(2)} + \lambda_{(N)} \geq \frac{2(\text{tr}(\mathbf{W}) - 1)}{N - 1} \quad (41)$$

where $\text{tr}(\cdot)$ denotes the trace of its argument.

Proof: Recall that the sum of eigenvalues of a matrix is equal to its trace:

$$\sum_{i=1}^N \lambda_i = \sum_{i=1}^N \lambda_{(i)} = \text{tr}(\mathbf{W}) \quad (42)$$

Noting that $\lambda_{(1)} = 1$ and rearranging the summation give

$$\sum_{i=2}^N \lambda_i = \text{tr}(\mathbf{W}) - 1. \quad (43)$$

Noting the fact that the eigenspectrum is convex function of the eigenvalue index, we can have:

$$\frac{(\lambda_{(2)} + \lambda_{(N)})(N - 1)}{2} \geq \sum_{i=2}^N \lambda_i \quad (44)$$

Substituting (43) into (44) results in the desired bound. ■

To avoid notational burden, let us denote the lowerbound for $\lambda_{(2)} + \lambda_{(N)}$ as following:

$$\xi \triangleq \frac{2(\text{tr}(\mathbf{W}) - 1)}{N - 1}. \quad (45)$$

Proposition 4, thus, provides an upperbound for mixing parameter α in terms of the trace of weight matrix \mathbf{W} satisfying the convergence and improvement factor conditions:

$$\alpha \leq \frac{\xi}{2 - \xi} \triangleq \Lambda(\xi). \quad (46)$$

If $\text{tr}(\mathbf{W})$ is calculated in a centralized fashion then it is a far less complicated operation than computing eigenvalues of the global weight matrix necessary to find optimum mixing parameter. Moreover, lower bound on the sum of the second largest and the smallest eigenvalue of matrix \mathbf{W} provided linearly depends on the average of the diagonal terms of the weight matrix and thus can also be calculated using standard consensus algorithm. However, convexity assumption made appears to be strong and unnecessary in many cases. Thus it is of interest to have a more general and rigorous approach to this problem based on fundamental properties of eigenspectrum of a matrix.

Upperbound provided above still requires the knowledge of the diagonal elements of \mathbf{W} . However, in the following we show that there exists an asymptotic upperbound $\Lambda_\infty(\xi)$ for α (satisfying the convergence and convergence improvements factor conditions) that does not require a global knowledge and can be calculated off-line. This quantity can be used in a fully local accelerated consensus. To be able to show this we need to make the following assumptions:

- (i) Sensors are randomly distributed in some area \mathcal{D} according to some known spatial distribution $p_{x,y}(x, y)$.
- (ii) Two nodes i and j in the sensor network are considered to be connected if they are closer to each other than

some predefined connectivity radius r_c :

$$\begin{cases} I\{r_{i,j}^2 \leq r_c^2\} = 1 & \text{if } r_{i,j}^2 \leq r_c^2 \\ I\{r_{i,j}^2 \leq r_c^2\} = 0 & \text{if } r_{i,j}^2 > r_c^2 \end{cases} \quad (47)$$

where $I\{r_{i,j}^2 \leq r_c^2\}$ is the indicator function and

$$r_{i,j}^2 = x_{i,j}^2 + y_{i,j}^2 \triangleq (x_i - x_j)^2 + (y_i - y_j)^2 \quad (48)$$

is the Euclidian distance between the nodes.

(iii) The weight matrix \mathbf{W} satisfying necessary and sufficient consensus convergence conditions is constructed according to some rule of the form:

$$\begin{cases} W_{ij} = I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j), & i \neq j \\ W_{ij} = 1 - \sum_{j=1, j \neq i}^N W_{ij}, & i = j \end{cases} \quad (49)$$

where $\mathcal{L}(d_i, d_j)$ is some function of the local connectivity degrees d_i and d_j of nodes i and j satisfying:

$$\begin{aligned} \sum_{j=1}^N \mathcal{L}(d_i, d_j) &= 1 \\ |\mathcal{L}(d_i, d_j)| &< 1 \end{aligned} \quad (50)$$

Given the assumptions outlined above, the following theorem provides an expression for an asymptotic upper bound on the value of mixing parameter α in terms of the expectation of $I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j)$.

Theorem 2: If \mathbf{W} is the $N \times N$ weight matrix constructed according to $\mathcal{L}(d_i, d_j)$ satisfying (50) and random variables ζ_i defined by:

$$\zeta_{i,N} = \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \quad (51)$$

are identically distributed with mean

$$|E\{\zeta_{i,N}\}| = |E\{\zeta\}| \quad (52)$$

and covariance structure satisfying

$$\sum_{N \in \mathbb{N}^+} \frac{\sigma_N}{N} \sqrt{R_{N-1}} + \frac{\sigma_N^2}{N^2} < \infty \quad (53)$$

where R_N and σ_N are defined as follows:

$$R_N \triangleq \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})(\zeta_{j,N} - E\{\zeta_{j,N}\})\} \quad (54)$$

$$\sigma_N \triangleq E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})^2\} \quad (55)$$

then the lower bound on $\lambda_{(2)} + \lambda_{(N)}$ given by Proposition 4 almost surely converges to

$$\xi_\infty \triangleq \lim_{N \rightarrow \infty} \xi \stackrel{a.s.}{\rightarrow} 2(1 - E\{\zeta\}) \quad (56)$$

and defines an asymptotic upper bound on α as $N \rightarrow \infty$ given by the following expression:

$$\alpha \stackrel{a.s.}{\leq}_{N \rightarrow \infty} \Lambda(\xi_\infty) \quad (57)$$

Proof: First, we note that by the construction of the weight matrix \mathbf{W} (49) we can transform the expression for ξ (41) as follows:

$$\xi = \frac{2}{N-1}(\text{tr}(\mathbf{W}) - 1) \quad (58)$$

$$= \frac{2}{N-1} \sum_{i=1}^N W_{ii} - \frac{2}{N-1} \quad (59)$$

$$= \frac{2N}{N-1} \frac{1}{N} \sum_{i=1}^N \left(1 - \sum_{j=1, j \neq i}^N W_{ij} \right) - \frac{2}{N-1} \quad (60)$$

$$= \frac{2(N-1)}{N-1} - \frac{2N}{N-1} \frac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \quad (61)$$

$$= 2 - \frac{2}{N-1} \sum_{i=1}^N \zeta_i \quad (62)$$

where to obtain the last equality we have used the definition (51). Note that $I\{r_{i,j}^2 \leq r_c^2\}$ is a random variable that is Bernoulli distributed according to a pmf:

$$\Pr\{I\{r_{i,j}^2 \leq r_c^2\}\} = p \quad (63)$$

$$\Pr\{I\{r_{i,j}^2 > r_c^2\}\} = 1 - p$$

In (63) p is the probability that two nodes in the sensor network are connected. This probability can be expressed as follows:

$$p = \iiint\limits_{\mathcal{S}} p_{x,y}(x_i, y_i) p_{x,y}(x_j, y_j) dx_i dy_i dx_j dy_j \quad (64)$$

where the set \mathcal{S} is defined over the area \mathcal{D} and connectivity region $r_{i,j}^2 \leq r_c^2$:

$$\mathcal{S} = \{(x_i, y_i, x_j, y_j) \mid (x_i - x_j)^2 + (y_i - y_j)^2 \leq r_c^2; x_i, y_i, x_j, y_j \in \mathcal{D}\} \quad (65)$$

The distribution of random variable $d = \mathcal{L}(d_i, d_j)$ is less straightforward to characterize as $\mathcal{L}(\cdot)$ is an arbitrary function and it is beyond the scope of current paper. We require, however, that $\mathcal{L}(\cdot)$ is such that (52) and (53) hold. It is straightforward to show that both mean and variance of random variables ζ_i are bounded under our

assumptions on $\mathcal{L}(d_i, d_j)$:

$$|E\{\zeta_i\}| = \left| E \left\{ \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right\} \right| \quad (66)$$

$$< \sup_{\mathcal{L}} \left| \sum_{j=1, j \neq i}^N \mathcal{L}(d_i, d_j) \right| = 2. \quad (67)$$

Moreover,

$$\sigma_N^2 = E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})^2\} \quad (68)$$

$$= E\{\zeta_{i,N}^2\} - E\{\zeta_{i,N}\}^2 \quad (69)$$

$$\leq E\{\zeta_{i,N}^2\} \quad (70)$$

$$= E \left\{ \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} \mathcal{L}(d_i, d_k) \right\} \quad (71)$$

$$< \sup_{\mathcal{L}} \left| \sum_{j=1, j \neq i}^N \mathcal{L}(d_i, d_j) \right| \sup_{\mathcal{L}} \left| \sum_{k=1, k \neq i}^N \mathcal{L}(d_i, d_k) \right| = 4. \quad (72)$$

Now, taking into account the results in (66) and (68) the following centered square integrable random process can be considered:

$$\chi_{i,N} = \zeta_{i,N} - E\{\zeta_{i,N}\}, i = 1 \dots N \quad (73)$$

We note that if the correlation function of this random process satisfies ergodicity assumptions implied by (53), we can invoke the Strong Law of Large Numbers stated by Poznyak [18] in Theorem 1 to show that

$$\frac{1}{N} \sum_{i=1}^N \chi_{i,N} \xrightarrow{a.s.} 0. \quad (74)$$

In its turn, this along with assumption (52) implies that

$$\frac{1}{N-1} \sum_{i=1}^N \zeta_{i,N} \xrightarrow{a.s.} E\{\zeta\}. \quad (75)$$

Thus combining the last result with (62) leads us to the following conclusion:

$$\xi_\infty = \lim_{N \rightarrow \infty} \left\{ 2 - \frac{2}{N-1} \sum_{i=1}^N \zeta_i \right\} \quad (76)$$

$$\xrightarrow[N \rightarrow \infty]{a.s.} 2(1 - E\{\zeta\}) \quad (77)$$

Finally, noting (46) concludes the proof. ■

Note the above result relies on the assumption that $\mathcal{L}(d_i, d_j)$ satisfies the conditions discussed above. The following remark shows that this assumption holds for the popular max-degree weight design scheme [5], [7]. The max-degree weights are very simple to compute and are well suited for distributed implementation. In particular, each node needs no information to determine the weights on its adjacent edges. Furthermore, the nodes do not need any global knowledge of the communication graph, or even the total number of nodes.

In the following, we demonstrate that the assumptions (52) and (53) hold in practice. Let $\mathcal{L}(d_i, d_j)$ be defined in accordance with maximum degree weight matrix construction mechanism:

$$\mathcal{L}(d_i, d_j) \triangleq \frac{1}{N} \quad (78)$$

In this case $\zeta_{i,N}$ takes the following form:

$$\zeta_{i,N} = \frac{1}{N} \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \quad (79)$$

Taking the expectation of $\zeta_{i,N}$ gives us:

$$E\{\zeta_{i,N}\} = \frac{1}{N} \sum_{j=1, j \neq i}^N E\{I\{r_{i,j}^2 \leq r_c^2\}\} = \frac{N-1}{N}p, \quad 0 \leq p \leq 1 \quad (80)$$

Now, consider the *double averaged* [18] correlation function (54) of the random process defined in (73)

$$\begin{aligned} R_N &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})(\zeta_{j,N} - E\{\zeta_{j,N}\})\} \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E \left\{ \left(\frac{1}{N} \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} - \frac{N-1}{N}p \right) \left(\frac{1}{N} \sum_{\ell=1, \ell \neq j}^N I\{r_{j,\ell}^2 \leq r_c^2\} - \frac{N-1}{N}p \right) \right\} \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E \left\{ \frac{1}{N^2} \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} \sum_{\ell=1, \ell \neq j}^N I\{r_{j,\ell}^2 \leq r_c^2\} \right\} \\ &\quad - \frac{N-1}{N^2}p \left(\sum_{k=1, k \neq i}^N E\{I\{r_{i,k}^2 \leq r_c^2\}\} + \sum_{\ell=1, \ell \neq j}^N E\{I\{r_{j,\ell}^2 \leq r_c^2\}\} \right) + \frac{(N-1)^2}{N^2}p^2 \\ &= \frac{1}{N^4} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1, k \neq i}^N \sum_{\ell=1, \ell \neq j}^N E\{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\} - 2\frac{(N-1)^2}{N^2}p^2 + \frac{(N-1)^2}{N^2}p^2 \\ &= \frac{1}{N^4} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1, k \neq i}^N \sum_{\ell=1, \ell \neq j}^N E\{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\} - \frac{(N-1)^2}{N^2}p^2 \end{aligned} \quad (81)$$

Let us look at the quadruple sum in (81). There are four possible cases to analyze:

- 1) $i = j$ and $k = \ell$

We note that the number of occurrences of this event is equal to $N(N-1)$. Furthermore, the expectation

under the sum can be easily evaluated as follows:

$$E \{ I\{r_{i,k}^2 \leq r_c^2\} \} = p^2 + p(1-p) = p \quad (82)$$

2) $i = j$ and $k \neq \ell$

We note that the number of occurrences of this event is equal to $N(N-1)(N-2)$. It is not necessary to evaluate the expectation directly. It is sufficient to note that this expectation corresponds to the probability of three arbitrary nodes in the network being connected. This probability is less than or equal to the probability of two arbitrary nodes being connected. Thus we have an obvious result:

$$E \{ I\{r_{i,k}^2 \leq r_c^2\} I\{r_{i,\ell}^2 \leq r_c^2\} \} = p^2 + p' \quad (83)$$

where p' is bounded by the following

$$0 \leq p' \leq p(1-p) \quad (84)$$

3) $i \neq j$ and $k = \ell$

We note that the number of occurrences of this event is also equal to $N(N-1)(N-2)$. Furthermore, the expectation in this case has the same meaning as (83)

$$E \{ I\{r_{i,k}^2 \leq r_c^2\} I\{r_{j,k}^2 \leq r_c^2\} \} = p^2 + p' \quad (85)$$

4) $i \neq j$ and $k \neq \ell$

We note that the number of occurrences of this event is equal to $N(N-1)(N^2-3N+3)$. Furthermore, the expectation is easy to evaluate using the independence of the random variables involved. We note that in this case the expectation corresponds to the probability of two independent randomly selected pairs of sensors in the network being connected:

$$E \{ I\{r_{i,k}^2 \leq r_c^2\} I\{r_{j,\ell}^2 \leq r_c^2\} \} = p^2 \quad (86)$$

Combining the result of the above analysis with (81) gives the following bound for the double averaged correlation function:

$$\begin{aligned}
R_N &= \frac{1}{N^4} (N(N-1)(p^2 + p(1-p)) + 2N(N-1)(N-2)(p^2 + p')) - \frac{(N-1)^2}{N^2} p^2 \\
&= \frac{(N-1)^2}{N^2} p^2 - \frac{(N-1)^2}{N^2} p^2 + \frac{N(N-1)}{N^4} p(1-p) + \frac{2N(N-1)(N-2)}{N^4} p' \\
&< \frac{p(1-p)}{N^2} + \frac{2p'}{N} \\
&< \frac{p(1-p)}{N^2} + \frac{2p(1-p)}{N} \\
&= p(1-p) \frac{2N+1}{N^2}
\end{aligned} \tag{87}$$

Now we can use (87) and (68) to show that series (53) converges. Indeed,

$$\sum_{N \in \mathbb{N}^+} \frac{\sigma_N}{N} \sqrt{R_{N-1}} + \frac{\sigma_N^2}{N^2} \tag{88}$$

$$< \sum_{N \in \mathbb{N}^+} \frac{2}{N} \sqrt{p(1-p) \frac{2(N-1)+1}{(N-1)^2}} + \frac{4}{N^2} \tag{89}$$

$$< \sum_{N \in \mathbb{N}^+} \frac{2\sqrt{p(1-p)(2N-1)}}{N(N-1)} + \frac{4}{N^2} \tag{90}$$

$$< \sum_{N \in \mathbb{N}^+} \frac{2\sqrt{2}\sqrt{p(1-p)}}{\sqrt{N}(N-1)} + \frac{4}{N^2} \tag{91}$$

it is obvious that series (91) converges, which implies the convergence of (88). Thus taking into account (62) and (79) the following result can be stated for the asymptotic lower bound on $\lambda_{(2)} + \lambda_{(N)}$ (56) in the situation when max-degree weights are used:

$$\xi_\infty^{MD} \xrightarrow[N \rightarrow \infty]{a.s.} 2(1-p) \tag{92}$$

Substituting (92) into (46) and rearranging the terms results in the following asymptotic upper bound on α :

$$\Lambda(\xi_\infty^{MD}) = \frac{1-p}{p}. \tag{93}$$

This result is due to the following fact. If we consider $I\{r_{i,j}^2 \leq r_c^2\}$ as noisy measurements bearing the information regarding the true value of p in the estimator of the form

$$\hat{p} = \frac{1}{N-1} \sum_{i=1}^N \zeta_{i,N} = \frac{1}{N} \sum_{i=1}^N \frac{1}{N-1} \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\}, \tag{94}$$

then according to our analysis of double averaged correlation function, the number of independent measurements grows at least $\mathcal{O}(\sqrt{N})$ times faster than the number of correlated ones. This fact ensures the ergodicity of the random process defined by $\zeta_{i,N}$, $i = 1 \dots N$. Hence the value of estimator \hat{p} (94) almost surely converges to the

true value of p as N goes to ∞ . We note that in max-degree setting p can be calculated as shown in Appendix IV.

IV. NUMERICAL EXAMPLES

First of all we would like to compare the convergence time results of the algorithm we propose with the algorithms presented in [8]. Let us introduce the following notation. MD and MH are the standard consensus algorithms based on maximum degree and Metropolis–Hastings weight matrices; OPT and BC are optimum and the best constant algorithms from [8]; MD–OM, MD–SM, and MD–SAM denote the proposed algorithm using maximum degree weight matrix and optimum α , suboptimum α chosen according to trace bound (46), and suboptimum α chosen according to asymptotic bound (46); MH–OM and MH–SM denote the proposed algorithm using Metropolis–Hastings weight matrix, optimum α and suboptimum α chosen according to trace bound (46). In this notation M is an integer number showing how many past samples are used in the predictor. The comparison results are presented in Fig. 1. Here the algorithm (9) is simulated in scenario that was analyzed analytically in section III, that is, in scenario when $M = 2$ and $k = 1$. It is interesting to note in Fig. 1(a) that in the case when our algorithm is used with maximum degree weight matrix and optimum α it gives almost exactly the same result as the best constant algorithm. It can also be seen in Fig. 1(a) that asymptotic upper bound on α holds when N is as low as 20 for the maximum degree weight matrix \mathbf{W} . The two curves corresponding to the bound based on the trace of weight matrix (46) and represented by \times and asymptotic upper bound developed in Theorem 2, (57) and represented by \circ are almost indistinguishable. At the same time, it is clear from Fig. 1(b) that although our algorithm is extremely simple and does not require any global optimization, it performs quite close to the optimum algorithm from [8] and outperforms the best constant algorithm when it is used in conjunction with Metropolis–Hastings weight matrix. Simulation results presented later in this section show that considerable further improvement is achievable using our algorithm with $M = 3$. From Fig. 1 we conclude that our algorithm best performs when it is used with Metropolis–Hastings weight matrix. Thus we conduct all the rest of the experiments using our algorithm with Metropolis–Hastings weight matrix only.

The plot of MSE for the proposed and standard consensus algorithms is shown in Fig. 2. We used the following simulation parameters to generate this figure. Number of trials is 500, number of nodes is 50 in Fig. 2(a) and 25 in Fig. 2(b), connectivity radius is equal to $\sqrt{\log N/N}$, mean value at every node is 1, and standard deviation of spatially white identically distributed Gaussian noise is 1. To obtain the results for our algorithm when $M = 3$ and $k = 1$ we used a 0.1 grid for the unknown parameter α and evaluated MSE of our algorithm at every value of α during each of the trials. After that we selected the lowest value of MSE that corresponded to the value of α closest to the optimum one. The conclusions that can be made by observing Fig. 2 are twofold. First, Fig. 2 confirms the fact that performance of our algorithm with $M = 2$ and $k = 1$ is very close to the optimum in terms of step-wise MSE decay. Second, our algorithm with $M = 3$ and $k = 1$ outperforms even the optimum algorithm

[8] in terms of step-wise MSE decay. This result can be attributed to the following fact. By properly combining several past local samples of consensus state with the current consensus states of the neighbors it is possible to incorporate into the next local consensus state estimate more information regarding the true consensus value than by using any proper combination of the current consensus states of the neighbors only.

Finally, we would like to compare the performance of our algorithm using different values of $M \geq 2$. The results of simulation are shown in Fig. 3. At this point we do not have expressions for optimum α in a more general case when the number samples M used by the predictor may be different from 2. Thus to generate this plot we used a 0.1 grid for the unknown parameter α and evaluated MSE of our algorithm at every value of k , M and α . After that for all k and M we selected the lowest value of MSE that was generated by the algorithm using α closest to the optimum one. These results were averaged over 500 trials. Fig. 3 implies that in our setting where parameters Θ are given by formula (99) the best performance is obtained if $M = 3$ is used. It can also be seen from this figure that the performance improvement is quite substantial. On the other hand, the performance of the algorithm degrades as M increases further. This can be explained by the fact that to predict the current state we use linear line parametrization for the predictor. As the number of samples in the predictor grows, linear parametrization becomes less and less accurate rendering the predictor to be less and less reliable. Also it is noteworthy that for $M = 3$ the results are not sensitive to change in predictor step k . In this case the difference in the performance of the algorithm with different k is negligible and can be attributed to the Monte–Carlo simulation error.

V. CONCLUDING REMARKS

APPENDIX I

GENERAL EXPRESSIONS FOR ARBITRARY M AND k

In this section we present the expressions for weights Θ in (17) and current state $\mathbf{x}(t)$ as a function of algorithm parameters and previous states for the case of arbitrary M and k . First, we note that we need an expression for the pseudoinverse \mathbf{A}^\dagger . By the definition of \mathbf{A} in (16) we get:

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & M \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ \cdots & \cdots \\ M & 1 \end{bmatrix} = \begin{bmatrix} 1 + 2 + \cdots + M - 1 + M & 1 + 1 + \cdots + 1 + 1 \\ 1^2 + 2^2 + \cdots + (M - 1)^2 + M^2 & 1 + 2 + \cdots + M - 1 + M \end{bmatrix} \quad (95)$$

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \frac{1}{2}(M + 1)^2 - \frac{1}{2}M - \frac{1}{2} & M \\ \frac{1}{3}(M + 1)^3 - \frac{1}{2}(M + 1)^2 + \frac{1}{6}M + \frac{1}{6} & \frac{1}{2}(M + 1)^2 - \frac{1}{2}M - \frac{1}{2} \end{bmatrix} \quad (96)$$

Next we note that the inverse of $\mathbf{A}^T \mathbf{A}$ can be found easily in closed form:

$$(\mathbf{A}^T \mathbf{A})^{-1} = \frac{2}{M(M-1)} \begin{bmatrix} -3 & \frac{6}{M+1} \\ 2M+1 & -3 \end{bmatrix} \quad (97)$$

Thus the expression for the pseudoinverse \mathbf{A}^\dagger follows immediately:

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \frac{2}{M(M-1)} \begin{bmatrix} \frac{6}{M+1} - 3 & \frac{12}{M+1} - 3 & \cdots & \frac{6M}{M+1} - 3 \\ 2M+1-3 & 2M+1-6 & \cdots & 2M+1-3M \end{bmatrix} \quad (98)$$

This results in the following expression for predictor weights:

$$\Theta = \mathbf{A}^{\dagger T} \mathbf{t}^{\text{P},k} = \frac{2}{M(M-1)} \begin{bmatrix} \left(\frac{6}{M+1} - 3\right)(M+k) + 2M+1-3 \\ \left(\frac{12}{M+1} - 3\right)(M+k) + 2M+1-6 \\ \dots \\ \left(\frac{6M}{M+1} - 3\right)(M+k) + 2M+1-3M \end{bmatrix} \quad (99)$$

Next, we present the general expression for $\mathbf{x}(t)$ as a function of algorithm parameters and previous states. As before, we assume that in general predictor weights are represented by the vector $\Theta = [\theta_1, \theta_2, \dots, \theta_M]$ given by the expression in (99). In this case the value of the predictor at time t is expressed as follows:

$$\mathbf{x}^{\text{P}}(t) = \sum_{j=1}^M \theta_j \mathbf{x}(t-M+j) \quad (100)$$

Recall that in our algorithm $\mathbf{x}(t)$ is a mixture of predicted state $\mathbf{x}^{\text{P}}(t)$ and the result of conventional consensus $\mathbf{x}^{\text{W}}(t)$ as in (9a)

$$\begin{aligned} \mathbf{x}(t) &= \alpha \mathbf{x}^{\text{P}}(t) + (1-\alpha) \mathbf{x}^{\text{W}}(t) \\ &= \alpha \sum_{j=1}^{M-2} \theta_j \mathbf{x}(t-M+j) + \alpha(\theta_{M-1} \mathbf{x}(t-1) + \theta_M \mathbf{x}^{\text{W}}(t)) + (1-\alpha) \mathbf{x}^{\text{W}}(t) \\ &= \alpha \sum_{j=1}^{M-2} \theta_j \mathbf{x}(t-M+j) + [\alpha \theta_{M-1} \mathbf{I} + (1-\alpha + \alpha \theta_M) \mathbf{W}] \mathbf{x}(t-1) \end{aligned} \quad (101)$$

The last expression on the right hand side of (101) can be easily recognized as a generalized version of (21). It can be seen from this expression that in the case of general M the output of our algorithm at time t cannot be written even in the generalized form of consensus $\mathbf{x}(t) = \mathbf{W}(t) \mathbf{x}(t-1)$, where $\mathbf{W}(t)$ is possibly time dependant weight matrix. Thus the optimization of the value of the parameter α using the expression (101) directly is impossible. This happens because of the presence of the terms under the sum in this equation. Each of the terms $\mathbf{x}(t-M+j)$ has expression similar to (101). The unfolding of the equation (101) back in time leads to very complicated expression

including the weighted sum of all the powers of \mathbf{W} from time 0 and up to time t . At this point in time we are not aware of how to deal with this extremely complex expression to optimize the value of α directly or through some suitable factorization.

APPENDIX II

ASYMPTOTIC CONVERGENCE CONDITIONS OF THE PREDICTOR BASED WEIGHT MATRIX

Note that to ensure asymptotical convergence, we need to prove the followings:

$$\mathbf{W}[\alpha]\mathbf{1} = \mathbf{1}, \mathbf{1}^T\mathbf{W}[\alpha] = \mathbf{1}^T, \rho(\mathbf{W}[\alpha] - \mathbf{J}) < 1 \quad (102)$$

where $\mathbf{J} = (N)^{-1}\mathbf{1}\mathbf{1}^T$. Substitution of (26) into the left part of (102) results in the following:

$$\mathbf{W}[\alpha]\mathbf{1} = ((1 + \alpha)\mathbf{W} - \alpha\mathbf{I})\mathbf{1} \quad (103)$$

$$= \mathbf{W}\mathbf{1} + \alpha\mathbf{W}\mathbf{1} - \alpha\mathbf{I}\mathbf{1} \quad (104)$$

$$= \mathbf{1} + \alpha\mathbf{1} - \alpha\mathbf{1} = \mathbf{1}. \quad (105)$$

Proof for the right part of (102) is analogous and is thus omitted.

We now consider the spectral radius of the predictor based weight matrix. It is sufficient to show that

$$\mathbf{W}[\alpha]\mathbf{u}_i = ((1 + \alpha)\lambda_{(i)} - \alpha)\mathbf{u}_i \quad (106)$$

for any α and $i = 1, 2, \dots, N$. Rearrangement of (26) gives the inverse relationship between \mathbf{W} and $\mathbf{W}[\alpha]$:

$$\mathbf{W} = (1 + \alpha)^{-1}(\mathbf{W}[\alpha] + \alpha\mathbf{I}) \quad (107)$$

Thus it follows from the definition of an eigenvector and the relationship between \mathbf{W} and $\mathbf{W}[\alpha]$ that for all α and i :

$$\mathbf{W}\mathbf{u}_i = \lambda_{(i)}\mathbf{u}_i$$

$$(1 + \alpha)^{-1}(\mathbf{W}[\alpha] + \alpha\mathbf{I})\mathbf{u}_i = \lambda_{(i)}\mathbf{u}_i \quad (108)$$

$$\mathbf{W}[\alpha]\mathbf{u}_i + \alpha\mathbf{I}\mathbf{u}_i = (1 + \alpha)\lambda_{(i)}\mathbf{u}_i \quad (109)$$

$$\mathbf{W}[\alpha]\mathbf{u}_i = ((1 + \alpha)\lambda_{(i)} - \alpha)\mathbf{u}_i.$$

Hence, $\lambda_{(1)}[\alpha] = 1$. Now, let us consider the spectral radius of $\mathbf{W}[\alpha] - \mathbf{J}$ defined as $\rho(\mathbf{W}[\alpha] - \mathbf{J})$:

$$\rho(\mathbf{W}[\alpha] - \mathbf{J}) \triangleq \max\{\lambda_{(2)}[\alpha], |\lambda_{(N)}[\alpha]|\}. \quad (110)$$

Note that for $\alpha > 0$, the eigenvalues experience a left shift since $\lambda_{(i)}[\alpha] = \alpha(\lambda_{(i)} - 1) + \lambda_{(i)}$ and $(\lambda_i - 1)$ is always

negative and $\alpha > 0$. Thus to ensure that $\rho(\mathbf{W} - \mathbf{J}) < 1$, we just need to make sure that $\lambda_{(N)}[\alpha] > -1$. Hence,

$$\alpha(\lambda_{(N)} - 1) - \alpha > -1 \Rightarrow \alpha < \frac{1 + \lambda_{(N)}}{1 - \lambda_{(N)}}. \quad (111)$$

Note that this is always satisfied for $\lambda_{(N)} \geq 0$ since $\alpha \in [0, 1]$.

APPENDIX III

OPTIMAL MIXING PARAMETER

In order to show that α^* is the optimal solution to the considered maximization problem, we need to show that $\alpha = \alpha^*$ is the global minimizer of $\rho(\mathbf{W}[\alpha] - \mathbf{J})$. Hence we define the following optimization problem:

$$\alpha^* = \arg \min_{\alpha} \rho(\mathbf{W}[\alpha] - \mathbf{J}) \quad (112)$$

$$= \arg \min_{\alpha} \max_{i \neq 1} (|\lambda_{(i)}[\alpha]|). \quad (113)$$

However, this problem can be converted into a simpler one:

$$\alpha^* = \arg \min_{\alpha} \max(|\lambda_{(N)}[\alpha]|, \lambda_{(2)}[\alpha]) \quad (114)$$

since $\lambda_{(N)}[\alpha]$ is the smallest and $\lambda_{(2)}\alpha$ is the largest eigenvalue of $(\mathbf{W}[\alpha] - \mathbf{J})$. Noting that

$$\max\{a, b\} = \frac{|a - b| + (a + b)}{2} \quad (115)$$

we can further simplify (114)

$$\alpha^* = \arg \min_{\alpha} (|\lambda_{(N)}[\alpha]| - \lambda_{(2)}[\alpha]) + (|\lambda_{(N)}[\alpha]| + \lambda_{(2)}[\alpha]) \quad (116)$$

$$\triangleq \arg \min_{\alpha} \mathcal{Q}[\alpha]. \quad (117)$$

Applying the result of the Proposition 1 reduces the objective function $G(\alpha)$ to the following:

$$\mathcal{Q}[\alpha] = ||\lambda_{(N)}[\alpha]| - \lambda_{(2)}[\alpha]| + (|\lambda_{(N)}[\alpha]| + \lambda_{(2)}[\alpha]) \quad (118)$$

$$= |(1 + \alpha)\lambda_{(N)} - \alpha| - ((1 + \alpha)\lambda_{(2)} - \alpha) + |(1 + \alpha)\lambda_{(N)} - \alpha| + ((1 + \alpha)\lambda_{(2)} - \alpha) \quad (119)$$

Note that

$$\lambda > \frac{\alpha}{1 + \alpha} \Rightarrow (1 + \alpha)\lambda - \alpha > 0, \text{ and, } \lambda < \frac{\alpha}{1 + \alpha} \Rightarrow (1 + \alpha)\lambda - \alpha < 0. \quad (120)$$

Then for $\alpha < \lambda_{(N)}/(1 - \lambda_{(N)})$, after some algebra, we obtain

$$\mathcal{Q}[\alpha] = |\alpha(\lambda_{(N)} - \lambda_{(2)}) - \lambda_{(2)} + \lambda_{(N)}| + \alpha(\lambda_{(2)} + \lambda_{(N)} - 2) + \lambda_{(2)} + \lambda_{(N)} \quad (121)$$

Note that $\mathcal{Q}'[\alpha]$ is discontinuous at $\alpha = -1$, given that $\alpha < \lambda_{(N)}/(1 - \lambda_{(N)})$. Consider the derivative of $\mathcal{Q}[\alpha]$ for $\alpha > -1$:

$$\frac{\partial \mathcal{Q}[\alpha]}{\partial \alpha} \Big|_{\alpha > -1} = \lambda_{(N)} - \lambda_{(2)} + \lambda_{(N)} + \lambda_{(2)} - 2 = 2(\lambda_{(N)} - 1) < 0 \quad (122)$$

since $|\lambda_{(N)}| < 1$. Although, we restricted ourselves to $0 \leq \alpha \leq 1$, for completeness, we consider the case where $\alpha < -1$,

$$\mathcal{Q}[\alpha] \Big|_{\alpha < -1} = \frac{\partial \mathcal{Q}[\alpha]}{\partial \alpha} G(\alpha) \Big|_{\alpha < -1} = -\lambda_{(N)} + \lambda_{(2)} + \lambda_{(N)} + \lambda_{(2)} - 2 = 2(\lambda_{(2)} - 1) < 0 \quad (123)$$

since $|\lambda_{(2)}| < 1$. Thus the objective function $\mathcal{Q}[\alpha]$ is strictly decreasing when $\alpha < \lambda_{(N)}/(1 - \lambda_{(N)})$. In the following, we show that, when $\alpha > \lambda_{(N)}/(1 - \lambda_{(N)})$, the objective function is strictly decreasing when $\alpha < \alpha^*$ and $\alpha > \alpha^*$, which would indicate that the $\alpha = \alpha^*$ is the global optimum value. Consider the following:

$$\mathcal{Q}[\alpha] = |\alpha(2 - \lambda_{(N)} - \lambda_{(2)}) - (\lambda_{(N)} + \lambda_{(2)})| + (\alpha + 1)(\lambda_{(2)} - \lambda_{(N)}) \quad (124)$$

Note that

$$\alpha = \alpha^* = \frac{\lambda_{(2)} + \lambda_{(N)}}{2 - \lambda_{(N)} - \lambda_{(2)}} \quad (125)$$

is a discontinuity point of $\mathcal{Q}'[\alpha]$ given that $\alpha > \lambda_{(N)}/(1 - \lambda_{(N)})$. Consider the derivative of $\mathcal{Q}[\alpha]$ for $\alpha > \alpha^*$:

$$\mathcal{Q}[\alpha] \Big|_{\alpha > \alpha^*} = 2 - \lambda_{(N)} - \lambda_{(2)} + \lambda_{(2)} - \lambda_{(N)} = 2(1 - \lambda_{(N)}) > 0 \quad (126)$$

since $|\lambda_{(N)}| < 1$. On the other hand, if $\alpha < \alpha^*$

$$\mathcal{Q}[\alpha] \Big|_{\alpha < \alpha^*} = -2 + \lambda_{(N)} + \lambda_{(2)} + \lambda_{(2)} - \lambda_{(N)} = 2(\lambda_{(2)} - 1) < 0 \quad (127)$$

since $|\lambda_{(2)}| < 1$.

To conclude the proof, we note that $\mathcal{Q}[\alpha]$ is strictly decreasing when $\alpha < \alpha^*$, and $\mathcal{Q}[\alpha]$ is strictly increasing when $\alpha > \alpha^*$, indicating that α^* is the global minimum of $\mathcal{Q}[\alpha]$ and hence $\rho(\mathbf{W}[\alpha] - \mathbf{J})$.

APPENDIX IV

PROBABILITY THAT TWO ARBITRARY NODES ARE CONNECTED

In this section we present the calculation of the probability that two randomly selected nodes in a sensor network with connectivity radius r_c and sensors uniformly distributed $p_{x_i, y_i}(x_i, y_i) = 1$, $x_i, y_i \in [0, 1]$ in a normalized square area \mathcal{D} such that $\mathcal{D} = \{x, y | x, y \in [0, 1]\}$ on the plane are connected. As was mentioned before, this

probability can be evaluated using integral of the form

$$p = \iiint\limits_S p_{x_i, y_i}(x_i, y_i) p_{x_j, y_j}(x_j, y_j) dx_i dy_i dx_j dy_j \quad (128)$$

where the set \mathcal{S} is defined as follows

$$\mathcal{S} = \{(x_i, y_i, x_j, y_j) \mid (x_i - x_j)^2 + (y_i - y_j)^2 \leq r_c^2; x_i, y_i, x_j, y_j \in [0, 1]\} \quad (129)$$

To facilitate calculation of the integral (128) given the set of integration limits (129) we can divide this problem into two parts: $r_c \leq 1$ and $1 < r_c \leq \sqrt{2}$. Note also that random variables x and y can be introduced:

$$x = x_i - x_j, \quad y = y_i - y_j. \quad (130)$$

It is obvious that due to the fact that $p_{x_i, y_i}(x_i, y_i)$ is uniform, the joint distribution of x, y is triangular:

$$f_{x,y}(x, y) = \begin{cases} \frac{1}{4}(1 - |x|)(1 - |y|) & \text{if } (x, y) \in [-1, 1] \\ 0 & \text{otherwise} \end{cases}. \quad (131)$$

Hence the integral in (128) can be reformulated into double integral:

$$p = \iint\limits_{\substack{\sqrt{x^2+y^2} < r_c, \\ (x,y) \in [-1,1]}} f_{x,y}(x, y) dx dy. \quad (132)$$

Due to the symmetry of the problem we can consider only a positive quadrant during the calculation of (132). In the case $r_c \leq 1$ it reduces to the following:

$$p = 4 \int_{x=0}^{r_c} (1-x) \int_{y=0}^{\sqrt{r_c^2-x^2}} (1-y) dy dx = \frac{1}{2}r_c^4 - \frac{8}{3}r_c^3 + \pi r_c^2. \quad (133)$$

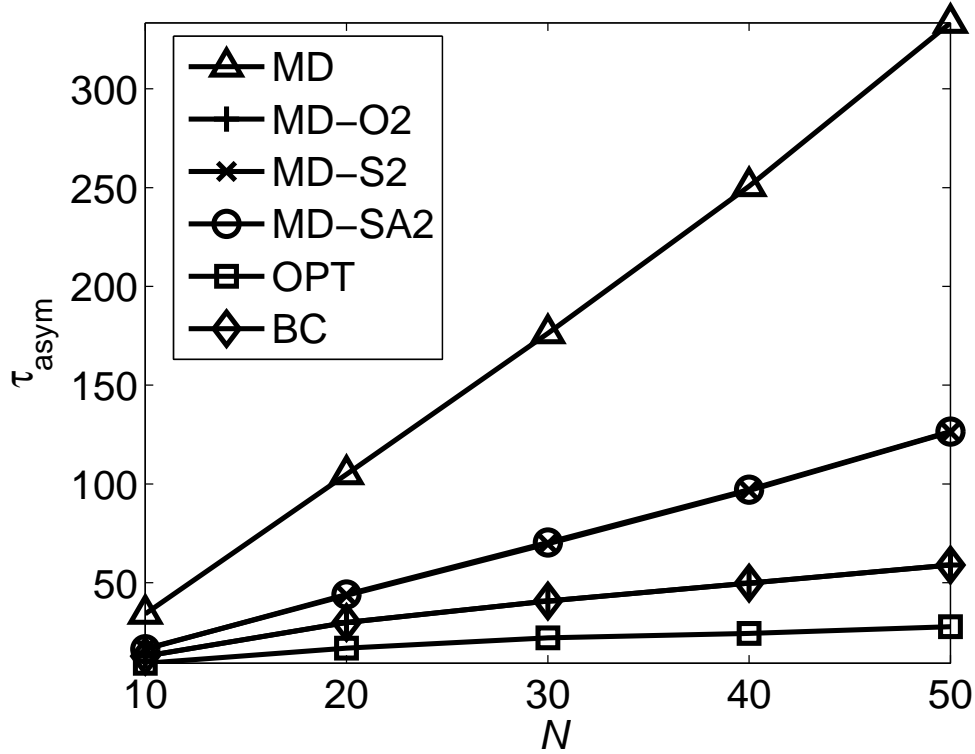
On the other hand, when $1 < r_c \leq \sqrt{2}$ we can reformulate (132) as follows:

$$\begin{aligned} p &= 1 - 4 \int_{y=\sqrt{r_c^2-1}}^1 (1-y) \int_{x=\sqrt{r_c^2-y^2}}^1 (1-x) dx dy \\ &= -\frac{1}{2}r_c^4 + \frac{8}{3}r_c^2 \sqrt{r_c^2-1} - 2r_c^2 + \frac{4}{3}\sqrt{r_c^2-1} - 2r_c^2 \arcsin\left(\sqrt{1-\frac{1}{r_c^2}}\right) + 2r_c^2 \arcsin\left(\frac{1}{r_c}\right) + \frac{1}{3}. \end{aligned} \quad (134)$$

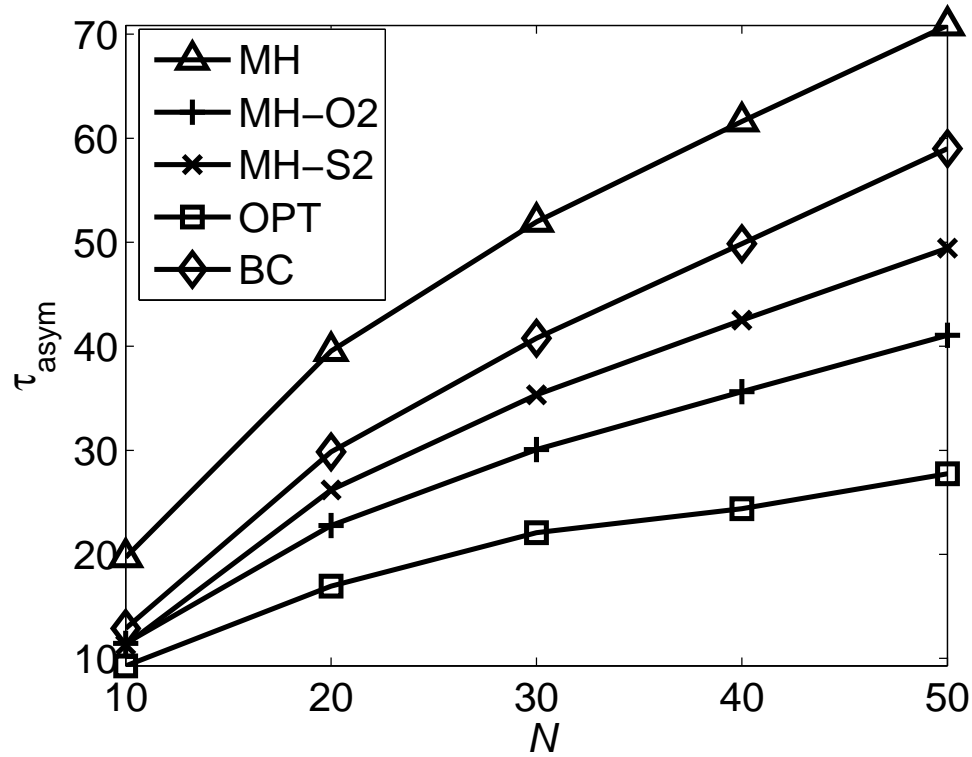
REFERENCES

- [1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

- [2] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proceedings of the International Workshop on Sensor Network Protocols and Applications*, 2003.
- [3] S. R. Madden, R. Szewczyk, M. J. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks," in *Proceedings of the Workshop on Mobile Computing Systems and Applications*, 2002.
- [4] A. Montresor, M. Jelasity, and O. Babaoglu, "Robust aggregation protocols for large-scale overlay networks," in *Proceedings of the International Conference on Dependable Systems and Networks*, 2004.
- [5] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proceedings of the IEEE/ACM Int. Symp. on Inf. Proc. in Sens. Netw.*, Los Angeles, CA, Apr. 2005.
- [6] C. C. Moallemi and B. V. Roy, "Consensus propagation," *IEEE Trans. Inform. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.
- [7] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, Jan. 2007.
- [8] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [9] N. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996.
- [10] C.-Z. Xu and F. Lau, *Load balancing in parallel computers: theory and practice*. Kluwer, Dordrecht, 1997.
- [11] Y. Rabani, A. Sinclair, and R. Wanka, "Local divergence of markov chains and the analysis of iterative load-balancing schemes," in *Proceedings of the IEEE Symp. on Found. of Comp. Sci.*, Palo Alto, CA, Nov. 1998.
- [12] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Automat. Contr.*, vol. 50, no. 5, pp. 655–661, 2005.
- [13] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 2004.
- [14] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Distributed sensor fusion using dynamic consensus," in *Proceedings of the 16th IFAC World Congress*, 2005.
- [15] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
- [16] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proceedings of the 36th ACM symposium on theory of computing*, 2004, pp. 561–568.
- [17] P. R. Kumar and P. Varaiya, *Stochastic systems: estimation, identification and adaptive control*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1986.
- [18] A. S. Poznyak, "A new version of the strong law of large numbers for dependent vector processes with decreasing correlation," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2000, pp. 2881–2882.



(a) Following algorithms were simulated. MD: \triangle , MD-O2: +, MD-S2: \times , MD-SA2: \circ , BC: \diamond , OPT: \square



(b) Following algorithms were simulated. MH: \triangle , MH-O2: +, MH-S2: \times , BC: \diamond , OPT: \square

Fig. 1. Asymptotic convergence time versus the number of nodes in the network

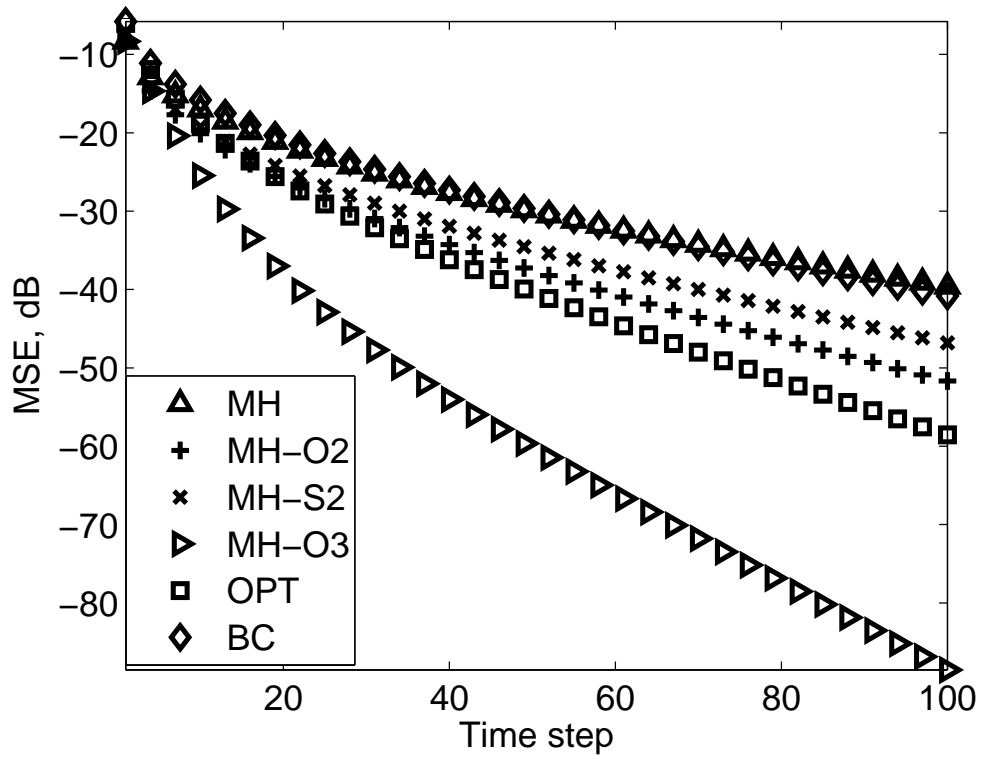
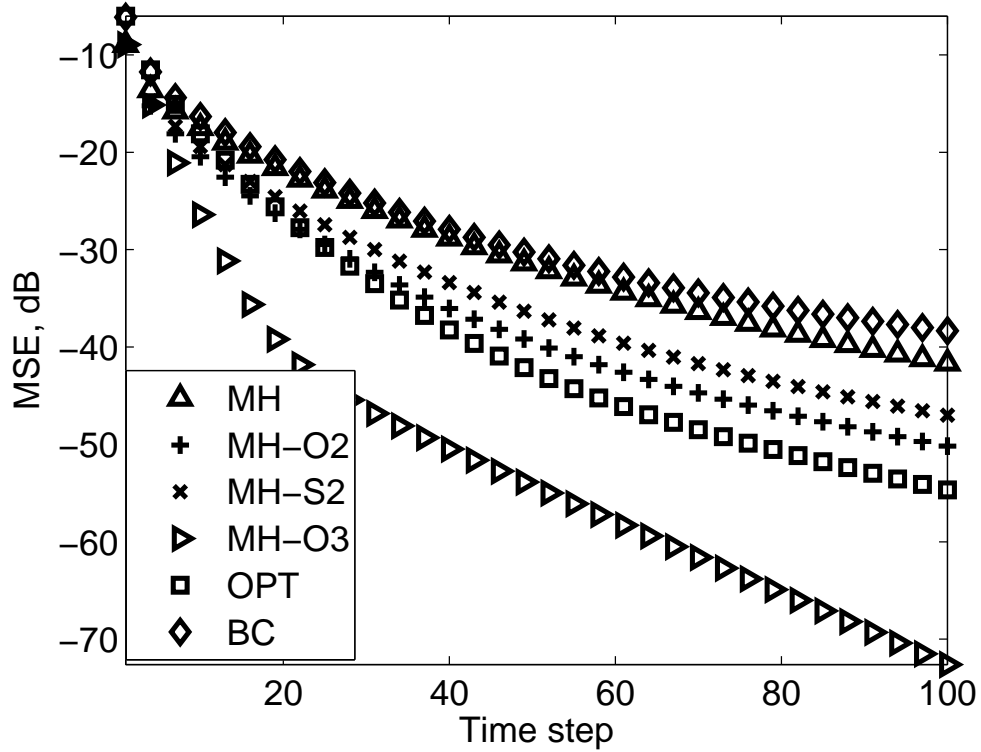
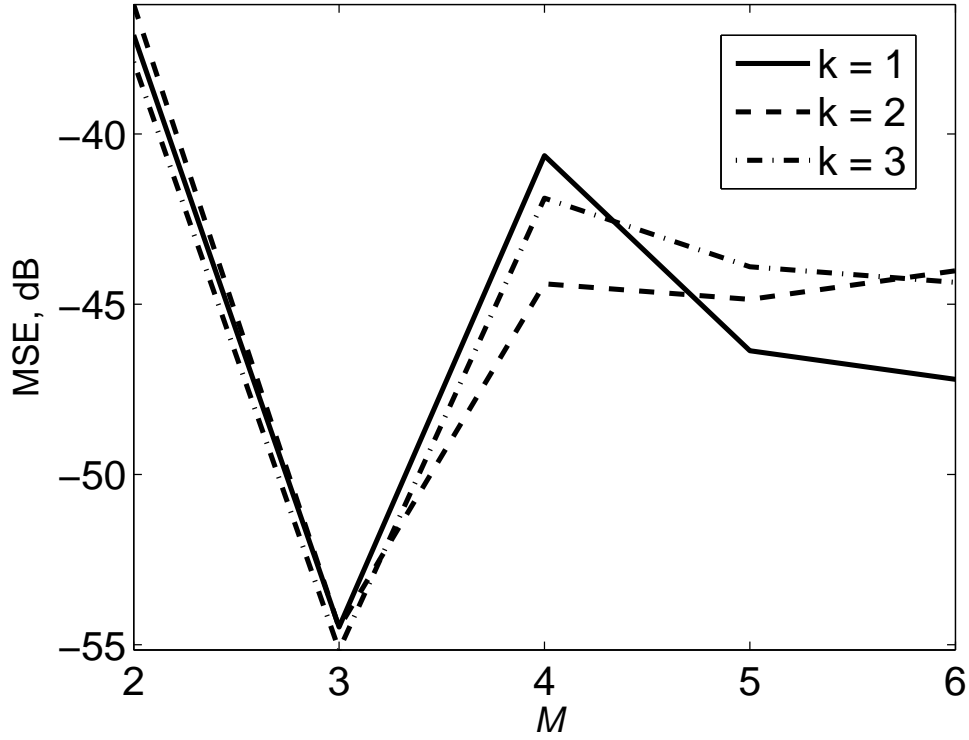
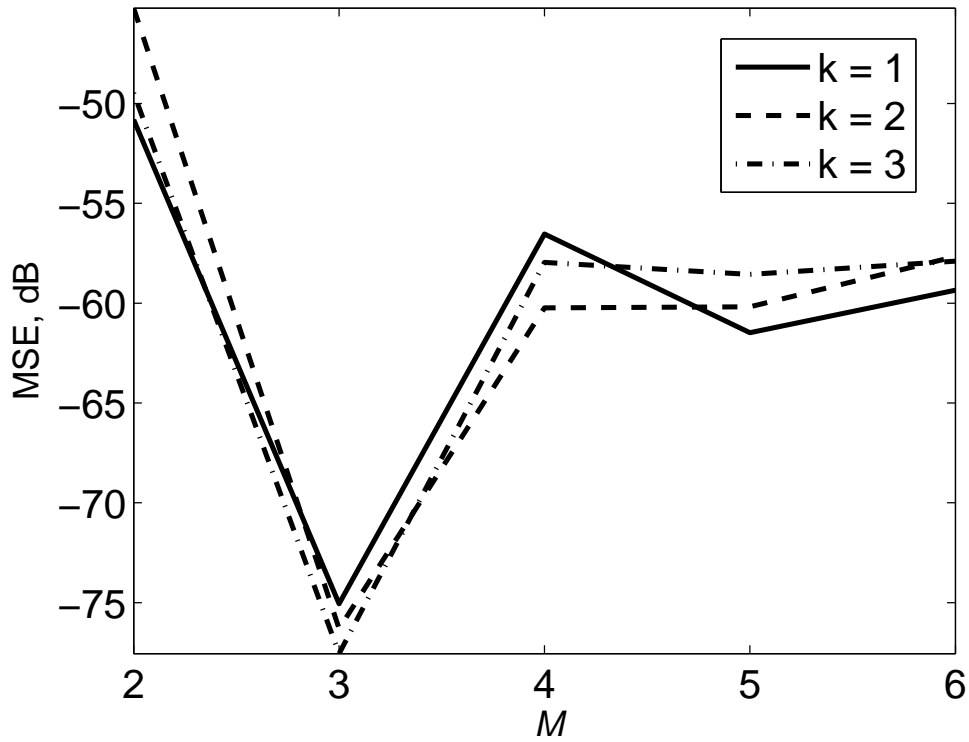
(a) Number of nodes $N = 25$ (b) Number of nodes $N = 50$

Fig. 2. MSE versus time step for the proposed and standard consensus algorithm. Following algorithms were simulated. MH: \triangle , MH-O2: $+$, MH-S2: \times , MH-O3: \triangleright , BC: \diamond , OPT: \square



(a) Number of iterations is 50, number of nodes is 50



(b) Number of iterations is 100, number of nodes is 50

Fig. 3. MSE versus the number of previous states M that were used to predict current local state for $k = 1, 2, 3$