

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

This paper proposes an approach to accelerate local, linear iterative network algorithms asymptotically achieving distributed average consensus. We focus on the class of algorithms in which each node initializes its “state value” to the local measurement and then at each iteration of the algorithm, updates this state value by adding a weighted sum of its own and its neighbors’ state values. Provided the weight matrix satisfies certain convergence conditions, the state values asymptotically converge to the average of the measurements, but the convergence is generally slow, impeding the practical application of these algorithms. In order to improve the rate of convergence, we propose a novel method where each node employs a linear predictor to predict future node values. The local update then becomes a convex (weighted) sum of the original consensus update and the prediction; convergence is faster because redundant states are bypassed. The method is linear and poses a small computational burden. For a concrete theoretical analysis, we prove the existence of a convergent solution in the general case and then focus on one-step prediction based on the current state, and derive the optimal mixing parameter in the convex sum for this case. Evaluation of the optimal mixing parameter requires knowledge of the eigenvalues of the weight matrix, so we present a bound on the optimal parameter. Calculation of this bound requires only local information. We provide simulation results that demonstrate the validity and effectiveness of the proposed scheme. The results indicate that the incorporation of a multi-step predictor can lead to convergence rates that are much faster than those achieved by an optimum weight matrix in the standard consensus framework.

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], [2], [3], [4]. Distributed average consensus is the task of calculating the average of a set of measurements made at different locations through the exchange of local messages. The goal is to avoid the need for complicated networks with routing protocols and topologies, but to ensure that the final average is available at every node. Distributed average consensus algorithms, which involve computations based only on local information, are attractive because they obviate the need for global communication and complicated routing, and they are robust to node and link failure. The roots of these algorithms can be traced back to the seminal work of Tsitsiklis [5], and there has been renewed interest because of their applicability in sensor networks [6], [7]. The algorithms can play an important role in agreement and synchronization tasks in ad-hoc networks [8] and also are good approaches for load balancing (with divisible tasks) in parallel computers [9], [10]. More recently, they have also been applied in distributed coordination of mobile autonomous agents [11] and distributed data fusion in sensor networks [12], [13], [14], [15].

In this paper we focus on a particular class of distributed iterative algorithms for average consensus: each node initializes its “state” to the local measurement, and then at each iteration of the algorithm, updates its state by adding a weighted sum of the local nodes [5], [12], [13], [16]. The algorithms in this class are time-independent, and the state values converge to the average of the measurements asymptotically [6]. The class is attractive because the algorithms are completely distributed and the computation at each node is very simple. The major deficiency is the relatively slow rate of convergence towards the average; often many iterations are required before the majority of nodes have a state value close to the average. In this paper, we address this deficiency, proposing a method that significantly improves the rate of convergence without sacrificing the linearity and simplicity.

A. Related Work

The convergence rate of distributed average consensus algorithms has been studied by several authors [6], [7]. Xiao, Boyd and their collaborators have been the main contributors of methods that strive to accelerate consensus algorithms through optimization of the weight matrix [6], [12], [17]. They showed that it is possible to formulate the problem of identifying the weight matrix that satisfies network topology constraints and minimizes the (worst-case) asymptotic convergence time as a convex semidefinite optimization task. This can be solved using a matrix optimization algorithm. Although elegant, the approach has two disadvantages. First, the convex optimization requires substantial computational resources and

can impose delays in configuration of the network. If the network topology changes over time, this can be of particular concern. Second, a straightforward implementation of the algorithm requires a fusion centre that is aware of the global network topology. In particular, in the case of on-line operation and a dynamic network topology, the fusion centre needs to re-calculate the optimal weight matrix every time the network topology changes. If such a fusion centre can be established in this situation, then the value of a consensus algorithm becomes questionable. To combat the second problem, Boyd *et al.* propose the use of iterative optimization based on the subgradient algorithm. Calculation of the subgradient requires knowledge of the eigenvector corresponding to the second largest eigenvalue of the weight matrix. In order to make the algorithm distributed, Boyd *et al.* employ decentralized orthogonal iterations [18], [17] for eigenvector calculation. The resulting algorithm, although distributed, is demanding in terms of time, computation and communication, because it essentially involves two consensus procedures.

Xiao and Boyd [6] also identify a suboptimal approach that leads to a much less demanding algorithm. In this approach, neighbouring edge weights are set to a constant, and the constant is optimized. The optimal constant is inversely proportional to the sum of the largest and the second smallest eigenvalues of the Laplacian spectrum. The calculation thus still requires knowledge of the connectivity pattern. The resultant weight matrix is called the ‘best constant’ weight matrix in [6]. The suboptimality of the best constant weight matrix stems from the fact that all the edge weights are constrained to be the same.

Also related to our proposed approach is the methodology proposed by Sundaram and Hadjicostis in [19]. Their algorithm achieves consensus in a finite number of time steps, and constitutes an optimal acceleration for some topologies. The disadvantage of the approach is that each node must know the complete weight matrix (and its powers), retain a history of all state values, and then solve a system of linear equations. Again, this disadvantage is most consequential in the scenario where nodes discover the network online and the topology is dynamic, so that the initialization operation must be performed frequently. However, even in the simpler case of a static topology, the overhead of distributing the required initialization information can diminish the benefits of the consensus algorithm unless it is performed many times. In contrast, our proposal maintains the local, sequential and linear properties of the standard consensus approach. Cao *et al.* propose an acceleration framework for gossip algorithms observing their similarity to the power method [20]. Their framework is based on the use of the weighted sum of shift registers storing the values of local gossip iterations. Although this framework is close to the one proposed in this paper, there are a number of points that make our proposal valuable. First, the approach in [20] utilizes the weight vector, but authors do not provide any solutions or directions for the weight vector design or optimization. Our approach, in its simplest form, optimizes a single parameter and we

demonstrate the optimal analytical solution for this parameter and the advantages our approach brings over the ones proposed in the literature. Second, the authors theoretically prove that if their algorithm converges then it converges to the true consensus value [20]. We provide a stronger theoretical guarantee: We prove that there always exists a set of algorithm parameters that leads to convergence to consensus. The idea of using higher order eigenvalue shaping filters has also appeared in [15], but the optimal choice of the filter parameters is still an open question.

B. Summary of Contributions

In this paper, we propose accelerating the convergence rate of a distributed average consensus algorithms by changing the state update to a convex combination of the standard consensus iteration and a linear prediction. We present a general framework and prove the existence of a solution, but focus on the special case of one-step prediction based only on the current state to gain further insight. For this case, we derive the optimal convex combination parameter, *i.e.*, the parameter which maximizes the worst-case asymptotic convergence rate. Since optimization of the parameter requires knowledge of the second largest and smallest eigenvalues of the weight matrix, we derive a suboptimal setting that requires much less information and is more readily implementable in a distributed setting. In the more general case of multi-step predictors, we generate bounds on the convex combination parameter that guarantee convergence of the consensus algorithm, but note that these are tighter than necessary. We report simulation results evaluating the behaviour of the optimal and suboptimal approaches. It also emerges that the proposed approach, when employing a multi-step predictor, has the potential to significantly outperform the standard consensus approach using the optimal weight matrix.

C. Paper Organization

The remainder 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. Section III describes the proposed algorithm and derives sufficient conditions on the mixing parameter for convergence to consensus. Section IV focuses on the special case of one-step prediction based only on the current state and provides an expression for the optimal mixing parameter. It also explores the improvement that is achieved in the convergence rate and describes a suboptimal, but effective and practical, approach for setting the mixing parameter. We report the results of numerical examples testing the proposed algorithms in Section V. Finally, Section VI concludes the paper.

II. PROBLEM FORMULATION

This section formulates the distributed average consensus task and briefly reviews the standard algorithms, as described in [5], [6]. We define a graph $G = (\mathcal{V}, \mathcal{E})$ as a 2-tuple, consisting of a set \mathcal{V} with $|\mathcal{V}| = N$ vertices, where $|\cdot|$ denotes the cardinality, and a set \mathcal{E} with $|\mathcal{E}|$ 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 also connected network topologies; the connectivity pattern of the graph is given by the $N \times N$ adjacency matrix $\Phi = [\Phi_{ij}]$, where

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

Denote the neighborhood of node i by $\mathcal{N}_i \triangleq \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$, and the degree of node i by $d_i \triangleq |\mathcal{N}_i|$.

We consider the 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 $\bar{x} \triangleq (N)^{-1} \mathbf{1}^T \mathbf{x}(0)$. In this paper we focus on a particular class of iterative algorithms for distributed average consensus. In this class, 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 [12], [13]. 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 necessary and sufficient conditions to ensure asymptotic average consensus [16]:

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

where \mathbf{J} is the averaging matrix

$$\mathbf{J} = \frac{1}{N} \mathbf{1}\mathbf{1}^T \quad (5)$$

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

$$\rho(\mathbf{W}) \triangleq \max_i \{|\lambda_i| : i = 1, 2, \dots, N\}. \quad (6)$$

Here $\{\lambda_i\}_{i=1}^N$ denote the eigenvalues of \mathbf{W} . Algorithms have been identified for generating weight matrices that satisfy the required convergence conditions if the underlying graph is connected, *e.g.*, Maximum-degree and Metropolis weights [16], [12].

In the next section, we describe our approach to accelerate the consensus algorithm. The approach is based on the observation that in the standard consensus procedure [6] the individual node state values converge in a smooth fashion. This suggests that it is possible to predict with good accuracy a future local node state based on past and current values. Combining such a prediction with the consensus operation thus has the potential to drive the overall system state closer to the true average at a faster rate than the standard consensus algorithm. Effectively, the procedure bypasses redundant states.

III. ACCELERATING DISTRIBUTED AVERAGE CONSENSUS FOR AN ARBITRARY WEIGHT MATRIX

This section describes the acceleration methodology. We first discuss the general form of the acceleration method. The primary parameter in the algorithm is the mixing parameter, α , which determines how much weight is given to the predictor and how much to the consensus operator. For the general case, we derive sufficient conditions on this parameter to ensure convergence to the average.

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. We are therefore motivated to use a linear predictor, thereby retaining the linear nature of the consensus algorithm.

In the proposed acceleration, we modify the state-update equations at a node to become a convex summation of the predictor and the value derived by application of the consensus weight matrix:

$$x_i(t) = \alpha x_i^P(t) + (1 - \alpha)x_i^W(t) \quad (7a)$$

$$x_i^W(t) = W_{ii}x_i(t-1) + \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t-1) \quad (7b)$$

$$x_i^P(t) = \theta_M x_i^W(t) + \sum_{j=1}^{M-1} \theta_j x_i(t-M+j) \quad (7c)$$

Here $\Theta_{M,k} = [\theta_1, \dots, \theta_M]$ is the vector of predictor coefficients.

The network-wide equations can then be expressed in matrix form by defining

$$\mathbf{W}_{[\alpha]}^{(M)} \triangleq (1 - \alpha + \alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I} \quad (8)$$

$$\mathbf{X}(t-1) \triangleq [\mathbf{x}(t-1)^T, \mathbf{x}(t-2)^T, \mathbf{x}(t-3)^T, \dots, \mathbf{x}(t-M+2)^T, \mathbf{x}(t-M+1)^T]^T \quad (9)$$

where \mathbf{I} is the identity matrix of the appropriate size and

$$\mathbf{W}_+ \triangleq \begin{bmatrix} \mathbf{W}_{[\alpha]}^{(M)} & \alpha\theta_{M-2}\mathbf{I} & \alpha\theta_{M-3}\mathbf{I} & \dots & \alpha\theta_2\mathbf{I} & \alpha\theta_1\mathbf{I} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix} \quad (10)$$

Here all the components of the block matrix are $N \times N$. We adopt the convention that for $t < 0$, $x(t) = x(0)$. The update equation is then simply $\mathbf{X}(t) = \mathbf{W}_+\mathbf{X}(t-1)$.

We adopt a time-invariant extrapolation procedure. The advantage of this approach is that the coefficients can be computed off-line as they do not depend on the data. We employ the best linear least-squares k -step predictor that extrapolates current state $x_i(t)$ of the i th node k time steps forward. Choosing higher k implies a more aggressive prediction component to the algorithm. The prediction coefficients become (see the detailed derivation in Appendix A):

$$\Theta_{M,k} = \mathbf{A}^{\dagger T} \mathbf{A}_k \quad (11)$$

where

$$\mathbf{A} \triangleq \begin{bmatrix} -M+1 & \dots & -1 & 0 \\ 1 & \dots & 1 & 1 \end{bmatrix}^T, \quad (12)$$

$\mathbf{A}_k \triangleq [k, 1]^T$ and \mathbf{A}^\dagger is the Moore-Penrose pseudoinverse of \mathbf{A} . Appendix A provides general expressions for the parameters $\Theta_{M,k}$.

It can be seen from (11) that $x_i^P(t)$ is a linear combination of M previous local consensus values. Thus the consensus acceleration mechanism outlined in equations (7a–7c) is fully local if it is possible to find an optimum value of α in (7a) that does not require any global knowledge.

B. Convergence of Predictor-based Consensus

In this section we provide a result that characterizes a range of α values that achieve convergence to the consensus for arbitrary, finite, values of M . Let $\lambda_{(i)}$ denote the i -th ranked eigenvalue. The main result is the following theorem:

Theorem 1. *If \mathbf{W} is symmetric, satisfies the conditions for asymptotic consensus (4), $\sum_{i=1}^M \theta_i = 1$, and*

$$0 \leq \alpha < \min \left(\frac{1 - |\lambda_{(N)}|}{-|\lambda_{(N)}| + |\theta_M||\lambda_{(N)}| + \sum_{j=1}^{M-1} |\theta_j|}, \frac{1 - |\lambda_{(2)}|}{-|\lambda_{(2)}| + |\theta_M||\lambda_{(2)}| + \sum_{j=1}^{M-1} |\theta_j|} \right) \quad (13)$$

then the general accelerated consensus algorithm achieves asymptotic convergence.

Proof: See Appendix B for a proof. ■

The first condition of the theorem is satisfied by the choice of predictor weights we have outlined, and the second condition specifies the bounds on the mixing parameter α . This is only a sufficient condition for convergence, but it does indicate that there is a range of values of α for every M that leads to asymptotic convergence. Significant improvements in the rate of convergence are generally achieved by α values outside the identified range due to the conservative nature of the proof.

IV. ONE-STEP PREDICTOR BASED DISTRIBUTED AVERAGE CONSENSUS

In order to better understand the algorithm, we analyze the important case when the algorithm (7) is based only on the current node states. For this case, we derive, in this section, the mixing parameter that leads to the optimal improvement of worst-case asymptotic convergence rate, and we characterize this improvement. Evaluating the optimal value requires knowledge of the second-largest and smallest eigenvalues, which can be difficult to determine. We therefore derive a bound on the optimal α value which requires less information; setting α to this bound results in close-to-optimal performance.

The predictor under consideration is a one-step extrapolator based on the current node state and the result of the standard consensus operator, *i.e.*, $k = 1$ and $M = 2$. In this case $\Theta_M = [-1, 2]^T$, so $x_i^P(t)$ can be expressed as follows:

$$x_i^P(t) = 2x_i^W(t) - x_i(t-1). \quad (14)$$

We can estimate the gradient of the state with respect to time as $\widehat{\nabla}x_i(t) \triangleq x_i^W(t) - x_i(t-1)$. Thus (14) can be rewritten as:

$$x_i^P(t) = x_i^W(t) + \widehat{\nabla}x_i(t). \quad (15)$$

The one-step predictor hence updates the current state in the gradient direction, to within estimation error.

Substituting (14) into (7a) we obtain the following expression for $x_i(t)$:

$$x_i(t) = \alpha(2x_i^W(t) - x_i(t-1)) + (1-\alpha)x_i^W(t) \quad (16)$$

$$= x_i(t-1)((1+\alpha)W_{ii} - \alpha) + (1+\alpha) \sum_{j \in \mathcal{N}_i} W_{ij}x_j(t-1). \quad (17)$$

This can be written in matrix form as:

$$\mathbf{x}(t) = \mathbf{W}[\alpha]\mathbf{x}(t-1) \quad (18)$$

where $\mathbf{W}[\alpha] = \mathbf{W}_{[\alpha]}^{(2)}$ is the weight matrix (as a function of α):

$$\mathbf{W}[\alpha] \triangleq (1 + \alpha)\mathbf{W} - \alpha\mathbf{I} \quad (19)$$

It is obvious from the previous equation that the predictor based weight matrix $\mathbf{W}[\alpha]$ has the same eigenvectors as \mathbf{W} and its eigenvalues are related to the eigenvalues of the original matrix \mathbf{W} via the relationship:

$$\lambda_{(i)}[\alpha] = (1 + \alpha)\lambda_{(i)} - \alpha \quad i = 1, \dots, N \quad (20)$$

The following proposition describes some properties of the weight matrix $\mathbf{W}[\alpha]$. We show that if the original weight matrix \mathbf{W} satisfies the conditions necessary for asymptotical convergence, then $\mathbf{W}[\alpha]$ also guarantees 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 $\mathbf{W}[\alpha]$. Then $\mathbf{W}[\alpha]$ satisfies the required convergence conditions if*

$$0 \leq \alpha < \frac{1 + \lambda_{(N)}}{1 - \lambda_{(N)}}. \quad (21)$$

Proof: See Appendix C. ■

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

$$\lambda_{(i)} \leq \lambda_{(j)} \Rightarrow \lambda_{(i)}[\alpha] \leq \lambda_{(j)}[\alpha] \quad (22)$$

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 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} , and their values are always smaller. Using this property, together with the definition of spectral radius (6), it is possible to formulate the problem of optimizing the mixing parameter to achieve the fastest asymptotic worst-case convergence rate as a convex optimization problem. In the following subsection, we outline this formulation and provide the closed-form solution.

A. Optimization of The Mixing Parameter

Recall that α is the mixing parameter that determines the relative influences of the standard consensus iteration and the predictor in (7a). In the following, we consider optimization of α to achieve the fastest

possible worst-case asymptotic convergence rate for the $M = 2, k = 1$ case of the accelerated consensus algorithm.

The spectral radius $\rho(\mathbf{W}[\alpha] - \mathbf{J})$ defines the worst-case measure of asymptotic convergence rate [16]. In fact, the spectral radius is directly related to the asymptotic convergence rate as defined in [6]:

$$r(\mathbf{W}) \triangleq \rho(\mathbf{W}[\alpha] - \mathbf{J}) = \sup_{\mathbf{x}(0) \neq \mathbf{J}\mathbf{x}(0)} \lim_{t \rightarrow \infty} \left(\frac{\|\mathbf{x}(t) - \mathbf{J}\mathbf{x}(0)\|}{\|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\|} \right)^{1/t}. \quad (23)$$

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:

$$\tau_{\text{asym}} \triangleq \frac{1}{\log(1/\rho(\mathbf{W}[\alpha] - \mathbf{J}))}. \quad (24)$$

Theorem 2. *The $M = 2, k = 1$ case of the proposed accelerated consensus algorithm has the fastest asymptotic worst-case convergence rate if the value of the mixing parameter α equals the following optimum value:*

$$\alpha^* = \frac{\lambda_{(N)} + \lambda_{(2)}}{2 - \lambda_{(N)} - \lambda_{(2)}} \quad (25)$$

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

Proof: See Appendix D. ■

As expected, the optimal mixing parameter α^* satisfies the following:

$$\alpha^* < \frac{1 + \lambda_{(N)}}{1 - \lambda_{(2)} + 1 - \lambda_{(N)}} \quad (26)$$

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

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

Remark 1. *Algebraic manipulations lead to the following equality:*

$$|\lambda_{(N)}[\alpha^*]| = \frac{\lambda_{(2)} - \lambda_{(N)}}{2 - \lambda_{(2)} - \lambda_{(N)}} = \lambda_{(2)}[\alpha^*]. \quad (28)$$

The optimal mixing parameter thus induces a shift in the eigenvalues so that the magnitudes of the second-largest and smallest eigenvalues of $\mathbf{W}[\alpha]$ are balanced. A similar effect is observed in the optimization conducted in [6]. It should be noted however, that even with the optimal choice of α the proposed algorithm for $M = 2$ case cannot outperform global optimization proposed in [6].

B. Convergence Rate Analysis

To see to what extent the proposed one-step extrapolation algorithm yields performance improvement over the conventional consensus procedure, we consider the ratio of the spectral radius of the corresponding matrices. This ratio gives the lower bound on performance improvement:

$$\gamma[\alpha] \triangleq \frac{\rho(\mathbf{W} - \mathbf{J})}{\rho(\mathbf{W}[\alpha] - \mathbf{J})} = \frac{\lambda_{(2)}}{\max\{\lambda_{(2)}[\alpha], |\lambda_{(N)}[\alpha]|\}} \quad (29)$$

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

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

$$\gamma[\alpha^*] = \frac{\lambda_{(2)}(2 - \lambda_{(2)} - \lambda_{(N)})}{\lambda_{(2)} - \lambda_{(N)}}. \quad (30)$$

Proof: Substituting α^* into (29) and taking into account the fact that $|\lambda_{(N)}[\alpha^*]| = \lambda_{(2)}[\alpha^*]$, after some algebraic manipulations, yield the expression for $\gamma[\alpha^*]$. ■

Although (25) provides an expression for the optimum mixing factor resulting in the 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 a suboptimum setting for α that results in less performance gain but requires considerably less information at each node.

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:*

$$0 < \alpha \leq \alpha^*. \quad (31)$$

Proof: The asymptotic worst-case convergence rate of algorithm (7) 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:

$$\begin{cases} \frac{\lambda_{(2)}(1 + \alpha) - \alpha}{\lambda_{(2)}} < 1 \\ \frac{\alpha(1 - \lambda_{(N)}) - \lambda_{(N)}}{\lambda_{(2)}} < 1 \end{cases} \quad (32)$$

indicating that

$$\begin{cases} \alpha(\lambda_{(2)} - 1) < 0 \\ \alpha < \frac{\lambda_{(2)} + \lambda_{(N)}}{1 - \lambda_{(N)}} \end{cases} \cdot \quad (33)$$

Observing that $\lambda_{(2)} - 1 < 0$, dividing the first part of (33) by $\lambda_{(2)} - 1$ and subtracting the same expression from the denominator of the second part we obtain the tightened version of (33):

$$0 < \alpha < \frac{\lambda_{(N)} + \lambda_{(2)}}{2 - \lambda_{(N)} - \lambda_{(2)}} \quad (34)$$

Finally, noting that the right hand side of this expression is equal to α^* concludes the proof. \blacksquare

We strive to identify a setting for α that guarantees an improvement in the convergence rate but does not require global knowledge of the weight matrix. Based on Proposition 3, if we lower-bound α^* , then setting α to this lower-bound will guarantee improvement in convergence rate. In order to lower-bound α^* , we need to lower-bound $\lambda_{(2)} + \lambda_{(N)}$. The next proposition provides such a bound in terms of the trace of the 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, namely,*

$$\lambda_{(i)} \leq \frac{N-i}{N-2} \lambda_{(2)} + \frac{i-2}{N-2} \lambda_{(N)}, \quad 2 \leq i \leq N \quad (35)$$

then

$$\lambda_{(2)} + \lambda_{(N)} \geq \xi \triangleq \frac{2(\text{tr}(\mathbf{W}) - 1)}{N - 1} \quad (36)$$

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:

$$\sum_{i=1}^N \lambda_i = \sum_{i=1}^N \lambda_{(i)} = \text{tr}(\mathbf{W}) \quad (37)$$

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

$$\sum_{i=2}^N \lambda_{(i)} = \text{tr}(\mathbf{W}) - 1. \quad (38)$$

Since, by assumption, the eigenspectrum is a convex function of the eigenvalue index, we have:

$$\frac{(\lambda_{(2)} + \lambda_{(N)})(N-1)}{2} \geq \sum_{i=2}^N \lambda_{(i)} \quad (39)$$

Substituting (38) into (39) results in the desired bound. \blacksquare

Proposition 4 leads to an upper bound for a setting of the mixing parameter α in order to achieve convergence at an improved rate:

$$\alpha \leq \frac{\xi}{2 - \xi} \triangleq \Lambda(\xi). \quad (40)$$

The advantage of this setting is that it is much simpler to calculate the trace $\text{tr}(\mathbf{W})$ in a distributed fashion than derive all of the eigenvalues, as required for determining the optimum mixing parameter. The lower bound depends linearly on the average of the diagonal terms of the matrix \mathbf{W} , which can be calculated using a standard consensus procedure. Although the result is useful and leads to a simpler mechanism for setting the mixing parameter, the convexity assumption is strong and is probably unnecessary for many topologies.

C. Random geometric graphs: choice of the mixing parameter

We now consider the special, but important, case of random geometric graphs, which can act as good topological models of wireless sensor networks, one of the promising application domains for consensus algorithms. For this case, we show that there exists an asymptotic upper bound $\Lambda(\xi_\infty)$ for α that can be calculated off-line. The random geometric graph is defined as follows: n nodes (vertices) are distributed in an area \mathcal{D} according to a point process with known spatial distribution $p_{x,y}(x,y)$. Two nodes i and j are connected, i.e. $\Phi_{ij} = 1$, if the Euclidean distance $r_{i,j}$ between them is less than some predefined connectivity radius r_c . The indicator function $I\{r_{i,j}^2 \leq r_c^2\} = 1$ whenever $r_{i,j}^2 \leq r_c^2$ holds.

We consider weight matrices \mathbf{W} constructed according to a rule of the following form:

$$\begin{cases} W_{ij} = I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j), & i \neq j \\ W_{ij} = 1 - \sum_{j=1, j \neq i}^N W_{ij}, & i = j \end{cases} \quad (41)$$

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:

$$\begin{aligned} \sum_{j=1}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) &= 1 \\ |\mathcal{L}(d_i, d_j)| &< 1 \end{aligned} \quad (42)$$

Let us introduce random variables $\zeta_{i,N}$ defined by:

$$\zeta_{i,N} = \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j). \quad (43)$$

Assume that $\mathcal{L}(d_i, d_j)$ is chosen so that these variables are identically distributed with mean $E\{\zeta_{i,N}\} = E\{\zeta\}$ and covariance structure satisfying

$$\sum_{N \in \mathbb{N}^+} \frac{\sigma_N}{N} \sqrt{R_{N-1}} + \frac{\sigma_N^2}{N^2} < \infty \quad (44)$$

where R_N and σ_N are defined as follows:

$$R_N \triangleq \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})(\zeta_{j,N} - E\{\zeta_{j,N}\})\} \quad (45)$$

$$\sigma_N^2 \triangleq E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})^2\} \quad (46)$$

For such a graph and weight matrix, the following theorem provides an asymptotic upper bound on the value of mixing parameter α in terms of the expectation $E\{\zeta\}$.

Theorem 3. *Let \mathbf{W} be the $N \times N$ weight matrix constructed according to (41). Suppose $\mathcal{L}(d_i, d_j)$ is chosen so that the random variables $\zeta_{i,N}$ defined by (43) are identically distributed with finite mean $E\{\zeta\}$ and covariance structure satisfying (44). Then the lower bound on $\lambda_{(2)} + \lambda_{(N)}$ given by Proposition 4 almost surely converges to*

$$\xi_\infty \triangleq \lim_{N \rightarrow \infty} \xi \xrightarrow[N \rightarrow \infty]{a.s.} 2(1 - E\{\zeta\}) \quad (47)$$

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

$$\alpha \stackrel{a.s.}{\leq}_{N \rightarrow \infty} \Lambda(\xi_\infty) \quad (48)$$

Proof: See Appendix E. ■

The above result relies on the assumption that $\mathcal{L}(d_i, d_j)$ satisfies the conditions discussed above. The following proposition states that this assumption holds for the popular max-degree weight design scheme [12], [16]. The max-degree weights are very simple to compute and are well suited for distributed implementation. In order to determine the weights, the nodes need no information beyond their number of neighbours.

Proposition 5. *If the weights in the weight matrix \mathbf{W} are determined using the max-degree weight approach, then assumptions of Theorem 3 hold and the asymptotic bound ξ_∞ on $\lambda_{(2)} + \lambda_{(N)}$ satisfies:*

$$\xi_\infty^{MD} \xrightarrow[N \rightarrow \infty]{a.s.} 2(1 - p) \quad (49)$$

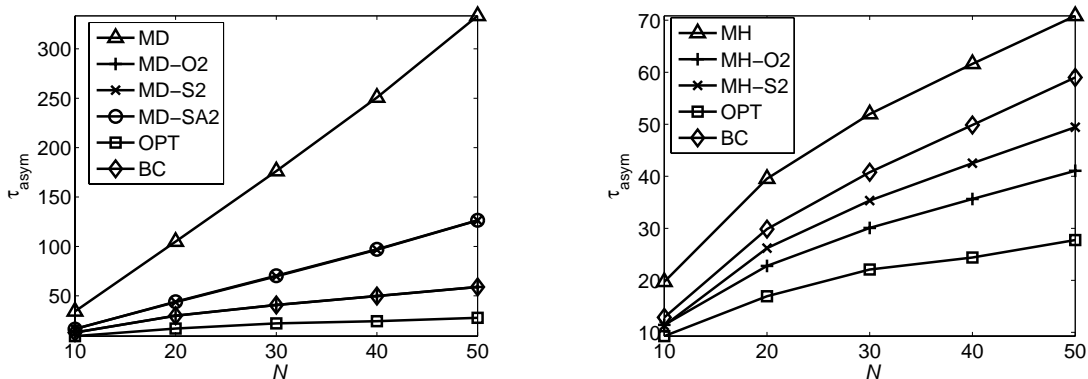
and

$$\Lambda(\xi_\infty^{MD}) = \frac{1 - p}{p}. \quad (50)$$

where p is the probability that two arbitrary nodes in a network are connected.

Proof: See Appendix F for the proof. ■

We note that p can be analytically derived for a given connectivity radius if the spatial distribution of the nodes is uniform [21]. Proposition 5 implies that for a random geometric graph with max-degree



(a) The convergence times for algorithms based on maximum-degree weight matrices as a function of the number of nodes in the network. Following algorithms were simulated. Standard Consensus (MD): \triangle ; Accelerated consensus with optimal α (MD-O2): $+$; Accelerated consensus with suboptimal α (MD-S2): \times ; Accelerated consensus with asymptotic suboptimal α (MD-SA2): \circ ; Best Constant [6] (BC): \diamond ; and Optimal Weight Matrix [6] (OPT): \square

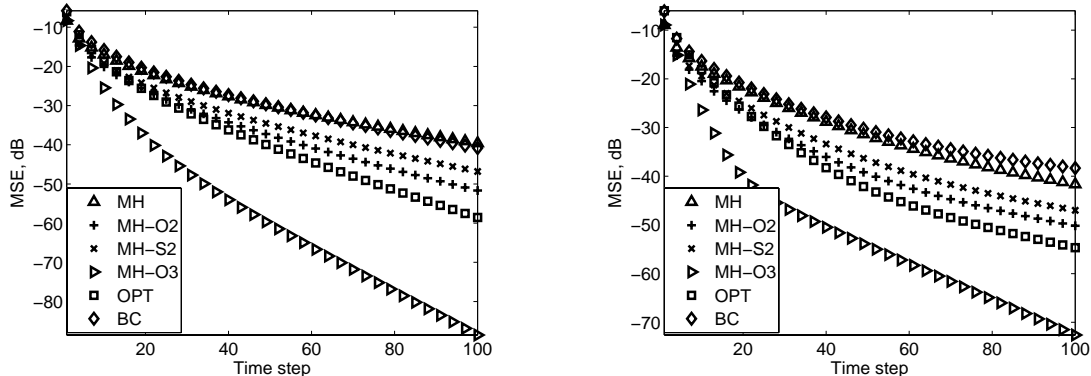
(b) The convergence times for algorithms based on Metropolis-Hastings weight matrices as a function of the number of nodes in the network. The following algorithms were simulated: Standard Consensus (MH): \triangle ; Accelerated consensus with optimal α (MH-O2): $+$; Accelerated consensus with suboptimal α (MH-S2): \times ; Best Constant [6] (BC): \diamond ; and Optimal Weight Matrix [6] (OPT): \square

Fig. 1. The asymptotic convergence time versus the number of nodes in the network. In (a), the standard and accelerated consensus algorithms are derived from the maximum-degree weight matrix; in (b) they are derived from the Metropolis-Hastings weight matrix.

weights α should be chosen to satisfy $\alpha \leq (1 - p)/p$. This result indicates that for highly connected graphs, which have a large value of p , a small α is desirable. For these graphs, standard consensus achieves fast mixing, so the prediction becomes less important and should be assigned less weight. In the case of a sparsely connected graph (small p), a large α is desirable. For these graphs, the convergence of standard consensus is slow because there are few connections, so the prediction component of the accelerated algorithm should receive more weight.

V. NUMERICAL EXAMPLES

In our simulation experiments, we consider a set of N nodes uniformly distributed on the unit square. The nodes establish bidirectional links to each other if the Euclidean distance between them is smaller than the connectivity radius, $\sqrt{\log N/N}$. Initial node measurements are generated as $\mathbf{x} = \theta + \mathbf{n}$ where $\theta = 1$ and \mathbf{n} is Gaussian distributed with $\sigma = 1$. Then, we regularize the data such that the average of all the values, \bar{x} , equals to 1. All simulation results are generated based on 500 trials (a different random



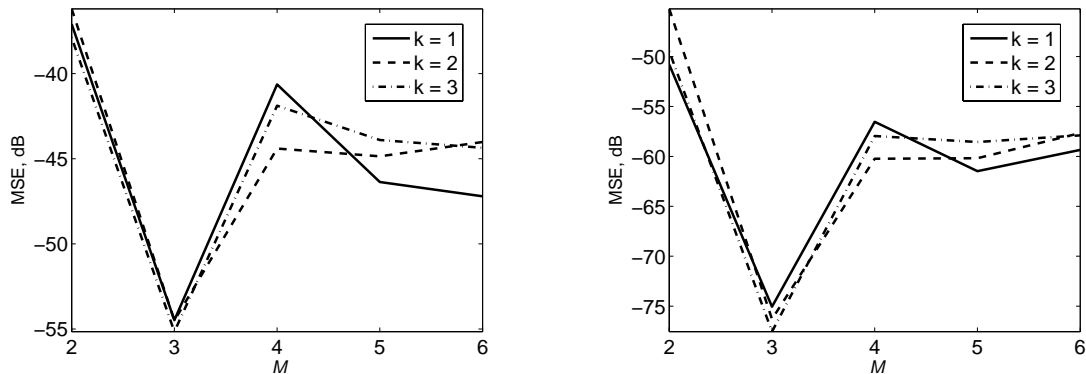
(a) MSE as a function of time step when the number of nodes $N = 25$

(b) MSE as a function of time step when the number of nodes $N = 50$

Fig. 2. Mean-squared-error (MSE) versus time step for the proposed and standard consensus algorithm. The upper panel depicts the results when the number of nodes in the network $N = 25$, and the lower panel depicts the results when $N = 50$. The following algorithms were simulated. Standard consensus (MH): \triangle ; Accelerated consensus, $M = 2$ with optimal α (MH-O2): $+$; Accelerated consensus $M = 2$ with suboptimal α (MH-S2): \times ; Accelerated consensus $M = 3$ (MH-O3): \triangleright ; Best Constant (BC): \diamond ; and optimal weight matrix (OPT): \square

graph is generated for each trial).

First, we compare the asymptotic convergence time (24) results of the algorithm we propose for the theoretically analyzed $M = 2$ and $k = 1$ case, against the algorithms presented in [6]. Fig. 1 compares the convergence times of the algorithms for the $M = 2$ and $k = 1$ case as a function of the number of nodes in the network. In Fig. 1(a), the maximum-degree weight matrix is used as the consensus operator for the standard and accelerated consensus algorithms; in Fig. 1(b), the Metropolis-Hastings weight matrix acts as the consensus operator. It is clear from Fig. 1 that although our algorithm is extremely simple and does not require any global optimization, it achieves performance improvements approaching those of the optimum algorithm from [6]. It outperforms the best constant algorithm when used in conjunction with the Metropolis-Hastings weight matrix. When Max-degree is utilized in the proposed algorithm, its asymptotic convergence time is very similar to that of the optimized best constant approach from [6] for the optimal choice of α . Fig. 1(a) also suggests that the asymptotic upper bound on α derived for a random geometric graph with maximum-degree weight matrix is applicable when N is as low as 20. The two curves corresponding to the bound based on the trace of weight matrix (40) (represented by \times) and the asymptotic upper bound developed in Proposition 5, (48) (represented by \circ) are almost indistinguishable.



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

Fig. 3. The mean-squared-error (MSE) at (a) iteration number 50, and (b) iteration number 100, as a function of the number of samples M used in the predictor. Results are depicted for one-, two- and three-step predictors ($k = 1, 2, 3$).

Since the performance of all algorithms is superior when the Metropolis–Hastings weight matrix is employed, the remainder of our simulations focus on this case. Fig. 2 shows the mean-squared-error (MSE) as a function of time for the standard, accelerated, best-constant and optimal weight matrix consensus algorithms. Three versions of the accelerated algorithm are depicted, including the $M = 2, k = 1$ case with optimal and suboptimal α , and the $M = 3, k = 1$ case. The number of nodes is 25 in Fig. 2(a) and 50 in Fig. 2(b). Since we have not developed a methodology for choosing an optimal setting for α when $M > 2$, we adopted the following procedure to choose α for the $M = 3$ and $k = 1$ case. For each trial, we evaluated the MSE for each value of α ranging from 0 to 1 at intervals of 0.1. We then chose the α that resulted in the lowest MSE for each trial at timestep 50. This represents a practically unachievable oracle α , so we stress that the purpose of the experiment is to illustrate the potential of the acceleration procedure. We do however observe that the random generation of the data has very little influence on the α value that is selected; it is the random graph and hence the initial \mathbf{W} matrix that governs the optimal value of α . This suggests that it is possible to develop a data-independent procedure to choose an optimal (or close-to-optimal) α value. Fig. 2 indicates that the accelerated consensus algorithm with $M = 2$ and $k = 1$ achieves step-wise MSE decay that is close to that obtained using the optimal weight matrix developed in [6]. The accelerated algorithm with $M = 3$ and $k = 1$ significantly outperforms the optimal weight matrix [6] in terms of step-wise MSE decay. The $M = 3$ case permits much more accurate prediction, which leads to the significant improvement in performance.

Finally, we compare the performance of our algorithm in terms of MSE decay for different values of k

and $M \geq 2$. The results of the simulation are shown in Fig. 3. At this point we do not have expressions for the optimum value of α in the more general case when $M > 2$. We use the same procedure as outlined in the previous experiment, evaluating the MSE for each possible value of α from 0 to 1 at intervals of 0.1 and selecting the one that achieves the minimum MSE. Fig. 3 implies that in our setting where parameters Θ are given by formula (60) the best performance is obtained if $M = 3$ is used and the performance difference is significant. Our choice of predictor exerts a significant influence here. We employ a linear parameterization that becomes less accurate as the number of samples employed in the predictor increases beyond 3. Note, however, that all of the depicted values of $M > 2$ achieve better performance than the optimal weight matrix. Although one might anticipate that there is an optimal k for each M that maximizes the convergence rate, our simulation experiments indicate that the value of k has little or no effect on convergence (see Fig. 3). Indeed, for the case $M = 2$, we can show analytically that the spectral radius of the equivalent weight matrix, which dictates the convergence rate, is independent of k [21].

VI. CONCLUDING REMARKS

We have presented a general, predictor-based framework to improve the rate of convergence of distributed average consensus algorithms. In contrast to previous acceleration approaches, the proposed algorithm is simple, fully linear, and parameters can be calculated offline. To gain further insight into the proposed algorithm, we focused on the special case of prediction based only on the current state, presenting theoretical and simulation results. In its most simple form, the proposed algorithm outperforms the optimal best constant algorithm from [6] and performs almost as well as the worst-case-optimal design algorithm of [6]. Simulation studies show that the proposed algorithm has the potential to outperform significantly the worst-case-optimal algorithm, if a suitable setting for the mixing parameter α can be identified. In future work, we plan to formulate the choice of α for the general case as an optimization problem, and examine the behaviour of the solutions. We hope that this will lead to practical schemes for setting α that guarantee convergence and achieve a convergence rate acceleration close to optimal.

APPENDIX A

GENERAL EXPRESSIONS FOR PREDICTOR WEIGHTS FOR ARBITRARY M AND k

In this appendix we present the expressions for the predictor weights $\Theta_{M,k}$ in (11) as a function of algorithm parameters and previous states for the case of arbitrary M and k .

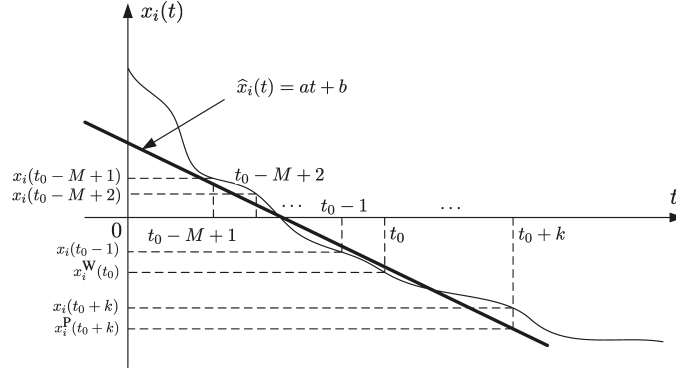


Fig. 4. Linear approximation to the model generating available data comprising linear predictor

First, we present the rationale behind the design of weights $\Theta_{M,k}$. As shown in Fig. 4, given the set of previous values at some time instant t_0 and node i , $x_i(t_0 - M + 1 : t_0) = [x_i(t_0 - M + 1), x_i(t_0 - M + 2), \dots, x_i(t_0 - 1), x_i^W(t_0)]^T$ we would like to design the best linear least squares approximation to the model generating the available data. Then using the approximate model we would like to extrapolate the current state k time steps forward. The easiest way to do this is to note that the approximate model of the form $\hat{x}_i(t) = at + b$ with a and b being the parameters of the linear model can be rewritten in the matrix form for the set of available data:

$$\hat{x}_i(t_0 - M + 1 : t_0) = \mathbf{A}_{t_0 - M + 1 : t_0} \psi. \quad (51)$$

Here

$$\mathbf{A}_{t_0 - M + 1 : t_0} = \begin{bmatrix} t_0 - M + 1 & 1 \\ t_0 - M + 2 & 1 \\ \dots & \dots \\ t_0 - 1 & 1 \\ t_0 & 1 \end{bmatrix} \quad \text{and} \quad \psi = [a, b]^T \quad (52)$$

Using the standard least squares technique we define the cost function

$$\mathcal{I}(\psi) = (x_i(t_0 - M + 1 : t_0) - \hat{x}_i(t_0 - M + 1 : t_0))^T (x_i(t_0 - M + 1 : t_0) - \hat{x}_i(t_0 - M + 1 : t_0)) \quad (53)$$

$$= (x_i(t_0 - M + 1 : t_0) - \mathbf{A}_{t_0 - M + 1 : t_0} \psi)^T (x_i(t_0 - M + 1 : t_0) - \mathbf{A}_{t_0 - M + 1 : t_0} \psi) \quad (54)$$

and find the optimal approximate linear model $\hat{\psi}$ as the global minimizer of the cost function:

$$\hat{\psi} = \arg \min_{\psi} \mathcal{I}(\psi) \quad (55)$$

Taking into account the convexity of the cost function and equating the derivative of $\mathcal{I}(\psi)$ with respect to ψ to zero we get the solution:

$$\widehat{\psi} = (\mathbf{A}_{t_0-M+1:t_0}^T \mathbf{A}_{t_0-M+1:t_0})^{-1} \mathbf{A}_{t_0-M+1:t_0}^T x_i(t_0 - M + 1 : t_0). \quad (56)$$

Now, given the linear approximation of the model generating current data, we extrapolate the current state k steps forward using $\mathbf{A}_{t_0+k} = [t_0 + k, 1]$:

$$x_i^P(t_0 + k) = \mathbf{A}_{t_0+k} \widehat{\psi} = \underbrace{\mathbf{A}_{t_0+k} (\mathbf{A}_{t_0-M+1:t_0}^T \mathbf{A}_{t_0-M+1:t_0})^{-1} \mathbf{A}_{t_0-M+1:t_0}^T}_{\Theta_{M,k}^T(t_0)} x_i(t_0 - M + 1 : t_0). \quad (57)$$

Finally, noting the time invariance of predictor weights $\Theta_{M,k}^T(t_0)$, that is $\Theta_{M,k}^T(t_0) = \Theta_{M,k}^T(0), \forall t_0$, we substitute $\mathbf{A}_{t_0-M+1:t_0}$ and \mathbf{A}_{t_0+k} by their time-invariant analogs \mathbf{A} and \mathbf{A}_k as defined in (11) and (12).

Second, we need an expression for the pseudoinverse \mathbf{A}^\dagger . From the definition of \mathbf{A} in (12) we can derive the inverse of $\mathbf{A}^T \mathbf{A}$ in closed form:

$$(\mathbf{A}^T \mathbf{A})^{-1} = \frac{2}{M(M+1)} \begin{bmatrix} \frac{6}{M-1} & 3 \\ 3 & 2M-1 \end{bmatrix} \quad (58)$$

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

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T = \frac{2}{M(M+1)} \begin{bmatrix} \frac{-6(M-1)}{M-1} + 3 & \frac{-6(M-2)}{M-1} + 3 & \cdots & 3 \\ -M+3-1 & -M+6-1 & \cdots & 2M-1 \end{bmatrix} \quad (59)$$

This results in the following expression for predictor weights:

$$\Theta_{M,k} = \mathbf{A}^{\dagger T} \mathbf{A}_{k+1} = \frac{2}{M(M+1)} \begin{bmatrix} \left(\frac{-6(M-1)}{M-1} + 3 \right) k - M + 3 - 1 \\ \left(\frac{-6(M-2)}{M-1} + 3 \right) k - M + 6 - 1 \\ \vdots \\ 3k + 2M - 1 \end{bmatrix} \quad (60)$$

APPENDIX B

PROOF OF THEOREM 1: EXISTENCE OF A CONVERGENT SOLUTION IN THE GENERAL CASE

This appendix presents a proof of Theorem 1. We commence by introducing an operator \mathbf{V} :

$$\mathbf{V} = \frac{\mathbf{W}_{[\alpha]}^{(M)}}{1 - \sum_{i=1}^{M-2} \alpha \theta_i} = \frac{(1 - \alpha + \alpha \theta_M) \mathbf{W} + \alpha \theta_{M-1} \mathbf{I}}{1 - \sum_{i=1}^{M-2} \alpha \theta_i} \quad (61)$$

Denoting $c_i = \alpha \theta_i$, we can write the first component of network-wide state recursion $\mathbf{X}(t) = \mathbf{W}^+ \mathbf{X}(t-1)$ as:

$$\mathbf{x}(t) = (1 - c_1 - c_2 - \dots - c_{M-2}) \mathbf{V} \mathbf{x}(t-1) + c_{M-2} \mathbf{x}(t-2) + \dots + c_1 \mathbf{x}(t-M+1) \quad (62)$$

where we set $\mathbf{x}(t) = \mathbf{x}(0)$ for any $t < 0$. Let us denote $S = (1 - c_1 - c_2 - \dots - c_{M-2})\|\mathbf{V} - \mathbf{J}\|$ and $\beta = S + |c_1| + |c_2| + \dots + |c_{M-2}|$. Here, as before, \mathbf{J} denotes the averaging operator. The following lemma provides the platform for the proof of the theorem, identifying sufficient conditions on α that guarantee $|\beta| < 1$.

Lemma 1. *If \mathbf{W} is symmetric, satisfies the conditions for asymptotic consensus (4), $\sum_{i=1}^M \theta_i = 1$, and*

$$0 \leq \alpha < \min \left(\frac{1 - |\lambda_{(N)}|}{-|\lambda_{(N)}| + |\theta_M||\lambda_{(N)}| + \sum_{j=1}^{M-1} |\theta_j|}, \frac{1 - |\lambda_{(2)}|}{-|\lambda_{(2)}| + |\theta_M||\lambda_{(2)}| + \sum_{j=1}^{M-1} |\theta_j|} \right) \quad (63)$$

then $|\beta| < 1$.

Proof: Using the triangle inequality and the definitions of S and \mathbf{V} , we can formulate a bound on $|\beta|$:

$$|\beta| = \left| (1 - c_1 - c_2 - \dots - c_{M-2})\|\mathbf{V} - \mathbf{J}\| + \alpha \sum_{j=1}^{M-2} |\theta_j| \right| \quad (64)$$

$$\leq \|(1 - \alpha + \alpha\theta_M)\mathbf{W} + \alpha\theta_{M-1}\mathbf{I} - \mathbf{J}\| + \alpha \sum_{j=1}^{M-2} |\theta_j|. \quad (65)$$

Thus if we ensure that this last expression is less than one, we guarantee that $|\beta| < 1$. We can reformulate this inequality using the symmetry of \mathbf{W} , \mathbf{I} , \mathbf{J} and the definition of the spectral radius of a matrix:

$$|(1 - \alpha + \alpha\theta_M)\lambda_{(i)} + \alpha\theta_{M-1}| + \alpha \sum_{j=1}^{M-2} |\theta_j| < 1, \quad \forall |\lambda_{(i)}| < 1 \quad (66)$$

Again applying the triangle inequality, we see that this relationship is satisfied if:

$$|1 - \alpha||\lambda_{(i)}| + \alpha \left(|\theta_M||\lambda_{(i)}| + \sum_{j=1}^{M-1} |\theta_j| \right) < 1, \quad \forall |\lambda_{(i)}| < 1 \quad (67)$$

Upon expansion of the modulus $|1 - \alpha|$, and with algebraic manipulation, we arrive at:

$$0 < \alpha < \mathcal{J}(\lambda_{(i)}) \triangleq \frac{1 - |\lambda_{(i)}|}{-|\lambda_{(i)}| + |\theta_M||\lambda_{(i)}| + \sum_{j=1}^{M-1} |\theta_j|}, \quad \forall |\lambda_{(i)}| < 1 \quad (68)$$

Now, let us examine the properties of the upper bound $\mathcal{J}(\lambda_{(i)})$ in (68). After some algebraic manipulations the derivative of $\mathcal{J}(\lambda_{(i)})$ for the two cases $\lambda_{(i)} > 0$ and $\lambda_{(i)} < 0$ takes the following form:

$$\begin{aligned} \frac{\partial}{\partial \lambda_{(i)}} \mathcal{J}(\lambda_{(i)} : \lambda_{(i)} > 0) &= \frac{1 - \sum_{j=1}^M |\theta_j|}{\left(-\lambda_{(i)} + |\theta_M||\lambda_{(i)}| + \sum_{j=1}^{M-1} |\theta_j|\right)^2} \\ \frac{\partial}{\partial \lambda_{(i)}} \mathcal{J}(\lambda_{(i)} : \lambda_{(i)} < 0) &= \frac{\sum_{j=1}^M |\theta_j| - 1}{\left(\lambda_{(i)} - |\theta_M||\lambda_{(i)}| + \sum_{j=1}^{M-1} |\theta_j|\right)^2} \end{aligned} \quad (69)$$

Taking into account the fact that $\sum_{j=1}^M |\theta_j| \geq 1$ we can make the following conclusion:

$$\begin{aligned} \frac{\partial}{\partial \lambda_{(i)}} \mathcal{J}(\lambda_{(i)} : \lambda_{(i)} > 0) &\leq 0, \quad \forall \lambda_{(i)} \\ \frac{\partial}{\partial \lambda_{(i)}} \mathcal{J}(\lambda_{(i)} : \lambda_{(i)} < 0) &\geq 0, \quad \forall \lambda_{(i)} \end{aligned} \quad (70)$$

Thus $\mathcal{J}(\lambda_{(i)})$ is nondecreasing when $\lambda_{(i)} < 0$ and nonincreasing when $\lambda_{(i)} > 0$. Hence if for any $\lambda_{(j)}$, $\lambda_{(k)}$, and $\lambda_{(i)}$ satisfying $|\lambda_{(j)}| > |\lambda_{(i)}|$ and $|\lambda_{(k)}| > |\lambda_{(i)}|$ there exists an α^* such that

$$0 < \alpha^* < \min(\mathcal{J}(\lambda_{(j)}), \mathcal{J}(\lambda_{(k)})) \quad (71)$$

then $0 < \alpha^* < \mathcal{J}(\lambda_{(i)})$ and (13) follows. To ensure that such α^* always exists for any $|\lambda_{(j)}| < 1$ and $|\lambda_{(k)}| < 1$ we note that $\mathcal{J}(\lambda_{(j)}) > 0, \quad \forall |\lambda_{(j)}| < 1$. This follows because $1 - |\lambda_{(j)}| > 0$. Moreover,

$$-|\lambda_{(i)}| + |\theta_M| |\lambda_{(i)}| + \sum_{j=1}^{M-1} |\theta_j| \geq -|\lambda_{(i)}| + |\lambda_{(i)}| \sum_{j=1}^M |\theta_j| \quad (72)$$

$$\geq -|\lambda_{(i)}| + |\lambda_{(i)}| = 0. \quad (73)$$

■

We now present the proof of Theorem 1.

Theorem 1: We first show that if the conditions of the theorem hold, then the average is preserved at each time step. To do this, it is necessary and sufficient to show that $\left(\left(1 - \sum_{i=1}^{M-2} c_i \right) \mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I} \right) \mathbf{1} = \mathbf{1}$ and $\mathbf{1}^T \left(\left(1 - \sum_{i=1}^{M-2} c_i \right) \mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I} \right) = \mathbf{1}^T$. We have:

$$\left(\left(1 - \sum_{i=1}^{M-2} c_i \right) \mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I} \right) \mathbf{1} = (1 - \alpha + \alpha \theta_M) \mathbf{W} \mathbf{1} + \alpha \theta_{M-1} \mathbf{I} \mathbf{1} + \alpha \sum_{i=1}^{M-2} \theta_i \mathbf{1} \quad (74)$$

$$= \left(1 - \alpha + \alpha \sum_{i=1}^M \theta_i \right) \mathbf{1} = \mathbf{1} \quad (75)$$

The proof of the condition $\mathbf{1}^T \left(\left(1 - \sum_{i=1}^{M-2} c_i \right) \mathbf{V} + \sum_{i=1}^{M-2} c_i \mathbf{I} \right) = \mathbf{1}^T$ is analogous and omitted.

We now show that $\mathbf{x}(t)$ converges to the average $\mathbf{J}\mathbf{x}(0)$. Our method of proof is induction. We show that $\|\mathbf{x}(t) - \mathbf{J}\mathbf{x}(0)\| \leq \beta^{\ell+1} \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\|$ where $\ell = \lfloor (t-1)/(M-1) \rfloor$. Lemma 1 implies that if the assumptions of the theorem are satisfied then $|\beta| < 1$, so the limit as t and consequently ℓ approaches infinity is 0. Initially, we show that the result holds for $\ell = 0$, or equivalently, $t = 1 \dots M-1$. We have,

using the triangle inequality and employing the result $\mathbf{V}\mathbf{J}\mathbf{x}(0) = \mathbf{J}\mathbf{x}(0)$:

$$\|\mathbf{x}(1) - \mathbf{J}\mathbf{x}(0)\| = \left\| \left(1 - \sum_{i=1}^{M-2} c_i\right) (\mathbf{V}\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)) + \sum_{i=1}^{M-2} c_i (\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)) \right\| \quad (76)$$

$$\leq \left(\left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| + \sum_{i=1}^{M-2} |c_i| \right) \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \quad (77)$$

$$= \beta \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \quad (78)$$

Similarly:

$$\|\mathbf{x}(2) - \mathbf{J}\mathbf{x}(0)\| = \left\| \left(1 - \sum_{i=1}^{M-2} c_i\right) (\mathbf{V}\mathbf{x}(1) - \mathbf{J}\mathbf{x}(0)) + \sum_{i=1}^{M-2} c_i (\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)) \right\| \quad (79)$$

$$\leq \left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| \|\mathbf{x}(1) - \mathbf{J}\mathbf{x}(0)\| + \sum_{i=1}^{M-2} |c_i| \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \quad (80)$$

$$\leq \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \left(\beta \left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| + \sum_{i=1}^{M-2} |c_i| \right) \quad (81)$$

$$\leq \beta \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\|, \quad (82)$$

where to obtain the last inequality we use the fact that $|\beta| < 1$. Using the same observations we can show the following for any n such that $2 < n < M$:

$$\|\mathbf{x}(n) - \mathbf{J}\mathbf{x}(0)\| = \left\| \left(1 - \sum_{i=1}^{M-2} c_i\right) (\mathbf{V}\mathbf{x}(n-1) - \mathbf{J}\mathbf{x}(0)) \right. \quad (83)$$

$$\left. + \sum_{i=2}^{n-1} c_{M-i} (\mathbf{x}(n-i) - \mathbf{J}\mathbf{x}(0)) + \sum_{i=n}^{M-1} c_{M-i} (\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)) \right\| \quad (84)$$

$$\leq \left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| \|\mathbf{x}(n-1) - \mathbf{J}\mathbf{x}(0)\| \quad (85)$$

$$+ \sum_{i=2}^{n-1} |c_{M-i}| \|\mathbf{x}(n-i) - \mathbf{J}\mathbf{x}(0)\| + \sum_{i=n}^{M-1} |c_{M-i}| \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \quad (86)$$

$$\leq \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \left(\beta \left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| + \beta \sum_{i=2}^{n-1} |c_{M-i}| + \sum_{i=n}^{M-1} |c_{M-i}| \right) \quad (87)$$

$$\leq \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\| \left(\left(1 - \sum_{i=1}^{M-2} c_i\right) \|\mathbf{V} - \mathbf{J}\| + \sum_{i=1}^{M-2} |c_i| \right) \quad (88)$$

$$\leq \beta \|\mathbf{x}(0) - \mathbf{J}\mathbf{x}(0)\|, \quad (89)$$

By almost identical manipulations, we can show that if the result holds for $\ell - 1$ and $t = (\ell - 1)(M - 1) + 1, \dots, (\ell - 1)(M - 1) + M - 1$, then it holds for ℓ and $t = \ell(M - 1) + 1, \dots, \ell(M - 1) + M - 1$.

APPENDIX C

PROOF OF PROPOSITION 1 : ASYMPTOTIC CONVERGENCE CONDITIONS OF THE PREDICTOR BASED WEIGHT MATRIX

Proof: In order to ensure asymptotical convergence, we need to prove the following properties:

$$\mathbf{W}[\alpha]\mathbf{1} = \mathbf{1}, \quad \mathbf{1}^T \mathbf{W}[\alpha] = \mathbf{1}^T, \quad \rho(\mathbf{W}[\alpha] - \mathbf{J}) < 1 \quad (90)$$

It is clear from (19) that $\mathbf{W}[\alpha]$ has the same eigenvectors as \mathbf{W} . Eigenvalues of $\mathbf{W}[\alpha]$ are connected to the eigenvalues via (20) and we conclude that $\lambda_{(1)}[\alpha] = 1$. Thus the two leftmost equations in (90) hold if \mathbf{W} satisfies asymptotic convergence conditions. Now, let us consider the spectral radius of $\mathbf{W}[\alpha] - \mathbf{J}$ defined as $\rho(\mathbf{W}[\alpha] - \mathbf{J})$:

$$\rho(\mathbf{W}[\alpha] - \mathbf{J}) \triangleq \max\{\lambda_{(2)}[\alpha], |\lambda_{(N)}[\alpha]|\}. \quad (91)$$

For $\alpha \geq 0$, the eigenvalues experience a left shift since $\lambda_{(i)}[\alpha] = \alpha(\lambda_{(i)} - 1) + \lambda_{(i)}$ and $(\lambda_i - 1)$ is always negative. It is also straightforward to see that $\lambda_{(i)} < \lambda_{(j)} \Rightarrow \lambda_{(i)}[\alpha] < \lambda_{(j)}[\alpha]$, $\forall i, j$. This implies that $\lambda_{(N)}[\alpha] < \lambda_{(2)}[\alpha] < 1$, so to ensure that $\rho(\mathbf{W} - \mathbf{J}) < 1$, we just need to make sure that $\lambda_{(N)}[\alpha] = \alpha(\lambda_{(N)} - 1) - \alpha > -1$. Rearrangement leads to $\alpha < \frac{1+\lambda_{(N)}}{1-\lambda_{(N)}}$, the condition expressed in (21). ■

APPENDIX D

PROOF OF THEOREM 2: OPTIMAL MIXING PARAMETER FOR THE ONE-STEP PREDICTOR MATRIX

Proof: 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:

$$\alpha^* = \arg \min_{\alpha} \rho(\mathbf{W}[\alpha] - \mathbf{J}) = \arg \min_{\alpha} \max_{i \neq 1} (|\lambda_{(i)}[\alpha]|). \quad (92)$$

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

$$\alpha^* = \arg \min_{\alpha} \max(|\lambda_{(N)} - \alpha(1 - \lambda_{(N)})|, |\lambda_{(2)} - \alpha(1 - \lambda_{(2)})|) \quad (93)$$

since $\lambda_{(N)}[\alpha]$ is the smallest and $\lambda_{(2)}[\alpha]$ is the largest eigenvalue of $(\mathbf{W}[\alpha] - \mathbf{J})$. Let us introduce $f(\alpha) = \lambda_{(N)} - \alpha(1 - \lambda_{(N)})$ and $g(\alpha) = \lambda_{(2)} - \alpha(1 - \lambda_{(2)})$. Clearly $|f(\alpha)|$ and $|g(\alpha)|$ are piecewise linear convex functions, with knots occurring where $f(\alpha) = 0$ and $g(\alpha) = 0$. Let these knots be $\alpha_f = \lambda_{(N)}/(1 - \lambda_{(N)})$ and $\alpha_g = \lambda_{(2)}/(1 - \lambda_{(2)})$. Since $\lambda_{(N)} < \lambda_{(2)}$ the magnitude of slope of $f(\alpha)$ exceeds that of $g(\alpha)$ and

$\alpha_f < \alpha_g$. Consider $h(\alpha) = \max(|f(\alpha)|, |g(\alpha)|)$, which is also piecewise linear and convex with knots α^- and α^+ occurring where $f(\alpha^-) = g(\alpha^-)$ and $f(\alpha^+) = -g(\alpha^+)$ respectively. Since $h(\alpha)$ is piecewise linear and convex with $h(\infty) = \infty$ and $h(-\infty) = \infty$ its global minimum occurs at one of the knots. It follows that the knots of $h(\alpha)$ satisfy $\alpha^- < \alpha_f < \alpha^+ < \alpha_g$. The fact that $|g(\alpha)|$ is decreasing if $\alpha < \alpha_g$ implies $g(\alpha^+) < g(\alpha^-)$. Hence the global minimum of $h(\alpha)$ occurs at $\alpha = \alpha^+$. Thus solving $f(\alpha^+) = -g(\alpha^+)$ for α^+ gives the solution for $\alpha^* = \alpha^+$. ■

APPENDIX E

RANDOM GEOMETRIC GRAPHS: PROOF OF THEOREM 3

Proof: By the construction of the weight matrix \mathbf{W} (41) we can transform the expression for ξ (36) as follows:

$$\xi = \frac{2}{N-1}(\text{tr}(\mathbf{W}) - 1) \quad (94)$$

$$= \frac{2N}{N-1} \frac{1}{N} \sum_{i=1}^N \left(1 - \sum_{j=1, j \neq i}^N W_{ij} \right) - \frac{2}{N-1} \quad (95)$$

$$= \frac{2(N-1)}{N-1} - \frac{2N}{N-1} \frac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \quad (96)$$

$$= 2 - \frac{2}{N-1} \sum_{i=1}^N \zeta_{i,N}. \quad (97)$$

In order to obtain the last equality we have used the definition (43). Note that $I\{r_{i,j}^2 \leq r_c^2\}$, the probability that two nodes are connected, is a Bernoulli random variable; denote the probability $\Pr\{I\{r_{i,j}^2 \leq r_c^2\} = 1\}$ by p . Note that an analytical expression for p can be derived if the nodes are uniformly distributed (see [21]); for other distributions, numerical integration can be employed to determine p .

We require that $\mathcal{L}(\cdot)$ is such that the $\zeta_{i,N}$ are identically distributed with finite mean and (44) holds. It is straightforward to show that both mean and variance of random variables $\zeta_{i,N}$ are bounded under our assumptions (42) on $\mathcal{L}(d_i, d_j)$:

$$|E\{\zeta_{i,N}\}| = \left| E \left\{ \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right\} \right| \quad (98)$$

$$\leq \sup \left| \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right| = 2. \quad (99)$$

The transition involves moving the expectation outside the modulus, replacing it by a supremum, and then application of the bounds in (42). Moreover,

$$\sigma_N^2 = E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})^2\} \quad (100)$$

$$\leq E\{\zeta_{i,N}^2\} \quad (101)$$

$$= E\left\{ \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} \mathcal{L}(d_i, d_k) \right\} \quad (102)$$

$$\leq \sup \left| \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \mathcal{L}(d_i, d_j) \right| \sup \left| \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} \mathcal{L}(d_i, d_k) \right| = 4. \quad (103)$$

Taking into account (98) and (100), we can consider the following centered square integrable random process:

$$\chi_{i,N} = \zeta_{i,N} - E\{\zeta_{i,N}\}, i = 1 \dots N \quad (104)$$

We note that if the correlation function of this random process satisfies ergodicity assumptions implied by (44), we can invoke the Strong Law of Large Numbers stated by Poznyak [22] (Theorem 1) to show that

$$\frac{1}{N} \sum_{i=1}^N \chi_{i,N} \xrightarrow{a.s.} 0. \quad (105)$$

In its turn, this along with the assumption $E\{\zeta_{i,N}\} = E\{\zeta\}$ implies that according to (104)

$$\frac{1}{N-1} \sum_{i=1}^N \zeta_{i,N} \xrightarrow{a.s.} E\{\zeta\}. \quad (106)$$

Combining (106) with (97) leads us to the following conclusion:

$$\xi_\infty = \lim_{N \rightarrow \infty} \left\{ 2 - \frac{2}{N-1} \sum_{i=1}^N \zeta_{i,N} \right\} \quad (107)$$

$$\xrightarrow[N \rightarrow \infty]{a.s.} 2(1 - E\{\zeta\}) \quad (108)$$

Finally, noting (40) concludes the proof. ■

APPENDIX F

RANDOM GEOMETRIC GRAPHS WITH MAXIMUM DEGREE WEIGHTS: PROOF OF PROPOSITION 5

Proof: Recall that the maximum degree weight design scheme employs the following settings: $\mathcal{L}(d_i, d_j) \triangleq N^{-1}$ for $\{i, j\} \in \mathcal{E}$ and $i \neq j$ and $W_{ii} = 1 - d_i/N$. With these choices, $\zeta_{i,N}$ takes the following form:

$$\zeta_{i,N} = \frac{1}{N} \sum_{j=1, j \neq i}^N I\{r_{i,j}^2 \leq r_c^2\} \quad (109)$$

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

$$E\{\zeta_{i,N}\} = \frac{1}{N} \sum_{j=1, j \neq i}^N E\{I\{r_{i,j}^2 \leq r_c^2\}\} = \frac{N-1}{N}p, \quad 0 \leq p \leq 1 \quad (110)$$

Now, consider the *double averaged* [22] correlation function (45) of the random process defined in (104)

$$\begin{aligned} R_N &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E\{(\zeta_{i,N} - E\{\zeta_{i,N}\})(\zeta_{j,N} - E\{\zeta_{j,N}\})\} \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E \left\{ \left(\frac{1}{N} \sum_{k=1, k \neq i}^N I\{r_{i,k}^2 \leq r_c^2\} - \frac{N-1}{N}p \right) \left(\frac{1}{N} \sum_{\ell=1, \ell \neq j}^N I\{r_{j,\ell}^2 \leq r_c^2\} - \frac{N-1}{N}p \right) \right\} \\ &= \frac{1}{N^4} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1, k \neq i}^N \sum_{\ell=1, \ell \neq j}^N E \{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\} - \frac{(N-1)^2}{N^2}p^2 \end{aligned} \quad (111)$$

Let us examine the quadruple sum in (111). There are four possible cases to analyze:

- 1) $i = j$ and $k = \ell$: The number of occurrences of this event is $N(N-1)$. The expectation can be evaluated:

$$E \{I\{r_{i,k}^2 \leq r_c^2\}^2\} = p^2 + p(1-p) = p \quad (112)$$

- 2) $i = j$ and $k \neq \ell$: 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. For some p' such that $0 \leq p' \leq p(1-p)$, we have:

$$E \{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{i,\ell}^2 \leq r_c^2\}\} = p^2 + p' \quad (113)$$

- 3) $i \neq j$ and $k = \ell$: This case is analogous to the preceding case.
- 4) $i \neq j$ and $k \neq \ell$: The number of occurrences of this event is equal to $N(N-1)(N^2 - 3N + 3)$. The expectation is easy to evaluate using the independence of the random variables involved. The expectation corresponds to the probability of two independent randomly selected pairs of nodes being connected:

$$E \{I\{r_{i,k}^2 \leq r_c^2\}I\{r_{j,\ell}^2 \leq r_c^2\}\} = p^2 \quad (114)$$

The above analysis leads to the following bound on the double averaged correlation function:

$$R_N < p(1-p) \frac{2N+1}{N^2} \quad (115)$$

Now we can use (115) and (100) to show that the series (44) converges. Indeed,

$$\sum_{N \in \mathbb{N}^+} \frac{\sigma_N}{N} \sqrt{R_{N-1}} + \frac{\sigma_N^2}{N^2} < \sum_{N \in \mathbb{N}^+} \frac{2\sqrt{2}\sqrt{p(1-p)}}{\sqrt{N}(N-1)} + \frac{4}{N^2}. \quad (116)$$

The series on the right hand side of (116) converges, which implies the convergence of the series in (44).

Since (44) is satisfied, we can apply Theorem 3 with (110) to derive (49). The result (50) follows immediately from the definition of $\Lambda(\xi)$ in (40). ■

REFERENCES

- [1] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom)*, Boston, MA, USA, Aug. 2000.
- [2] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proc. Int. Workshop on Sensor Network Protocols and Applications*, Seattle, WA, USA, May 2003.
- [3] S. Madden, R. Szewczyk, M. Franklin, and D. Culler, "Supporting aggregate queries over ad-hoc wireless sensor networks," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Callicoon, NY, USA, June 2002.
- [4] A. Montresor, M. Jelasity, and O. Babaoglu, "Robust aggregation protocols for large-scale overlay networks," in *Proc. Int. Conf. Dependable Systems and Networks*, Florence, Italy, June 2004.
- [5] J. Tsitsiklis, "Problems in decentralized decision making and computation," Ph.D. dissertation, Massachusetts Institute of Technology, Nov. 1984.
- [6] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.
- [7] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.
- [8] N. Lynch, *Distributed Algorithms*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996.
- [9] C.-Z. Xu and F. Lau, *Load balancing in parallel computers: theory and practice*. Kluwer, Dordrecht, 1997.
- [10] Y. Rabani, A. Sinclair, and R. Wanka, "Local divergence of Markov chains and the analysis of iterative load-balancing schemes," in *Proc. IEEE Symp. on Foundations of Computer Science*, Palo Alto, CA, Nov. 1998.
- [11] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [12] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IEEE/ACM Int. Symp. on Information Processing in Sensor Networks*, Los Angeles, CA, Apr. 2005.
- [13] C. Moallemi and B. Roy, "Consensus propagation," *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.
- [14] D. Spanos, R. Olfati-Saber, and R. Murray, "Distributed sensor fusion using dynamic consensus," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, July 2005.
- [15] D. Scherber and H. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proc. ACM/IEEE Int. Symp. Information Processing in Sensor Networks*, Berkeley, CA, USA, Apr. 2004.
- [16] 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.

- [17] 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.
- [18] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. ACM Symp. Theory of Computing*, Chicago, IL, USA, June 2004.
- [19] S. Sundaram and C. Hadjicostis, "Distributed consensus and linear function calculation in networks: An observability perspective," in *Proc. IEEE/ACM Int. Symp. Information Proc. in Sensor Networks*, Cambridge, MA, USA, Apr. 2007.
- [20] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *Proc. 44th Annual Allerton Conf. Communication, Control, and Computation*, Monticello, IL, USA, Sep. 2006.
- [21] T. Aysal, B. Oreshkin, and M. Coates, "Accelerated distributed average consensus via localized node state prediction," Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada, Tech. Rep., Oct. 2007, available at <http://www.tsp.ece.mcgill.ca/Networks/publications-techreport.html>.
- [22] A. Poznyak, "A new version of the strong law of large numbers for dependent vector processes with decreasing correlation," in *Proc. IEEE Conf. Decision and Control*, Dec. 2000.