Locally Constructed Algorithms for Distributed Computations in Ad-Hoc Networks *

Dzulkifli S. Scherber
Electrical and Computer Engineering &
Institute for Systems Research
University of Maryland
College Park, MD USA
dzuls@glue.umd.edu

Haralabos C. Papadopoulos
Electrical and Computer Engineering &
Institute for Systems Research
University of Maryland
College Park, MD USA
babis@glue.umd.edu

ABSTRACT

In this paper we develop algorithms for distributed computation of a broad range of estimation and detection tasks over networks with arbitrary but fixed connectivity. The distributed algorithms we develop are linear dynamical systems that generate sequences of approximations to the desired computation. The algorithms are locally constructed at each node by exploiting only locally available and macroscopic information about the network topology. We present methods for designing these distributed algorithms so as to optimize the convergence rates to the desired computation and demonstrate their performance characteristics in the context of a problem of signal estimation from multi-node signal observations in Gaussian noise.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*

General Terms

Algorithms

Keywords

Sensor networks, distributed algorithms, distributed estimation

1. INTRODUCTION

Ad-hoc networks of autonomous sensors and actuators are attractive solutions for a broad range of data collecting applications due to their inherent mobility, the spatial flexibility they allow in collecting measurements, and, consequently the quality of the data they can provide. Such networks find use in various civilian and military applications, including target tracking and surveillance for robot navigation [11, 16] source localization [13] and radar applications [19], data gathering for weather forecasting and environmental applications[5, 7, 12], and medical monitoring and imaging [2, 3, 14]. In general, the networks envisioned for many of these applications involve large numbers of possibly randomly distributed inexpensive sensor nodes, with limited sensing, processing, and communication power on board. A typical ad-hoc network is shown in Fig. 1. Scalability, robustness to changes in network topology, and fault tolerance are few of the challenges encountered in performing these tasks in ad-hoc sensor networks [1].

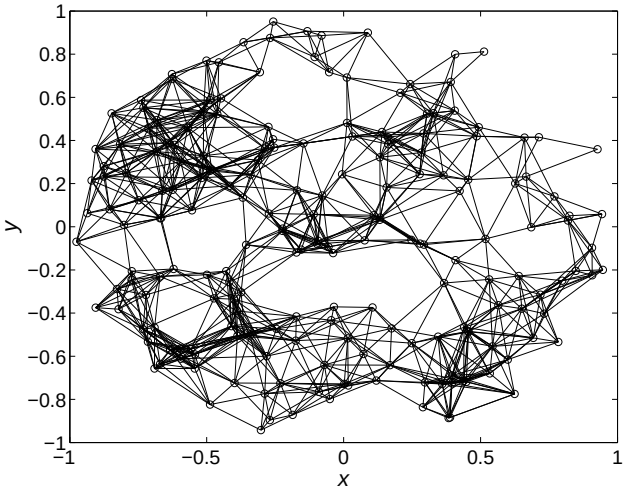

Figure 1: Ad-hoc network with 200 nodes.

As technological advances allow the deployment of growing numbers of increasingly powerful sensors, many important constraints invariably arise [1]. In many of these applications, limitations in bandwidth and sensor battery power and computing resources place tight constraints in the rate and form of information that can be exchanged. Equally important, growing network sizes together with changes in the network topology due to node mobility and/or node failures make global knowledge of the changing network topology impractical [15]. As a result, decentralized approaches are becoming increasingly attractive over their centralized counterparts for performing data fusion in large ad-hoc networks of sensors and actuators.

*This work was partly supported by the US Army Research Lab CTA under Award No DAAD19-01-2-0008, and an NSF CAREER under Award No. CCR0093137.

IPSN'04, April 26–27, 2004, Berkeley, California, USA.

Distributed routing and fusion algorithms have received attention recently in an effort to design algorithms that are scalable, fault-tolerant and robust to changes in the network topology. These algorithms are constructed via locally available information for routing and fusion. In particular, localized techniques have been developed for routing information in these networks [15, 17]. These techniques mainly focus on dynamic localized routing of important distributed information based on information-driven sensor querying and data processing [10, 20]. These algorithms for localized routing can be applied to networks with arbitrary connectivity, but the single fusion center they employ remains a single point of failure in the network [17, 6, 18].

An alternative approach to such distributed implementations involves viewing each node in the network as a fusion host in its own right and developing integrated routing/fusion algorithms such that the fusion objectives are made available to every node in the network. Such distributed algorithms for data fusion over ad-hoc networks are considered in [6], with objective to develop fully distributed Kalman filter implementations (with no global fusion center) over arbitrary topologies, by assuming only local knowledge of the network topology. The investigation is based on a model where each node in the network uses local connectivity information to perform partial data fusion and to share results with its neighbors. The inherent challenge is identifying and managing redundant information due to communication loops. Based on the analysis in [6], it is concluded that, in general, such distributed implementations cannot be used for arbitrary fusion tasks over arbitrary topologies.

The subject of this paper is the development of algorithms for ad-hoc sensor networks that allow distributed computation of a class of functions of the node data. These algorithms generate at each node in the network a sequence of estimate approximations to the desired computation. As we demonstrate, by exploiting knowledge of local network connectivity, these algorithms can be locally constructed and optimized at each node so that the sequences they generate converge to the desired global computation objective.

The locally constructed algorithms we develop for performing global computations have strong connections to large systems of weakly coupled nonlinear oscillators. Such systems have been exploited for pattern recognition [8] and have served as mathematical models used to explain how biological species achieve global synchronization via local interactions [4]. As we demonstrate, the two key properties of reciprocity and balancing found in these large systems of locally coupled nonlinear oscillators, are also present in the locally constructed distributed algorithms we develop for performing global computations.

The outline of the paper is as follows. In Sec. 2 we present the class of distributed rules of interest and describe the sense in which algorithmic approximation to the desired computation is evaluated. In Sec. 3 we develop distributed fusion rules for computing global averages. In Sec. 4 we consider these algorithms in the context of a problem of linear estimation in Gaussian noise, and present metrics for quantifying the performance of these algorithms. In Sec. 5 we present an evaluation of these algorithms based on a representative set of Monte-Carlo simulations. Sec. 6 includes a brief discussion on generalizations of these algorithms for computation of a broader class of tasks. Finally, Sec. 7 contains some concluding remarks.

2. SYSTEM MODEL AND DEFINITIONS

We consider a network of N nodes that wish to compute a scalar function of their data, *i.e.*, they wish to compute $G(\mathbf{x})$ where

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T , \tag{1}$$

and x_i denotes the scalar observation at the ith sensor node. The fusion rules we consider in this paper are algorithms that are implemented on the given network topology and generate at each node a sequence of approximations to $G(\mathbf{x})$.

We assume network topologies according to which, each node can establish bidirectional noise-free communication with a subset of the nodes in the network. We represent an N-node topology by the $N \times N$ matrix $\Phi = \{\phi_{ij}\}_{i,j \in \{1,\, 2,\, \cdots,\, N\}}$, where for $i \neq j$

$$\phi_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ directly communicate} \\ 0 & \text{otherwise} \end{cases} . \tag{2a}$$

For convenience, we set

$$\phi_{ii} = -\sum_{j \neq i} \phi_{ij} \ . \tag{2b}$$

As a result, $|\phi_{ii}|$ corresponds to the number of nodes in direct communication with the ith node. Throughout we assume *connected topologies*, *i.e.*, topologies for which there exists a multi-hop communication path between every pair of nodes in the network. For connected topologies, it is evident that $|\phi_{ii}| > 0$ for all i. Furthermore, Φ is negative semidefinite and has one eigenvalue equal to 0, and $N-1$ negative eigenvalues. We also let $\mathcal{U}_j$ denote the set of nodes that have a direct (bidirectional) communication link with node j, *i.e.*,

$$\mathcal{U}_j \stackrel{\triangle}{=} \{i \in \{1,\, 2\, \cdots,\, N\}\ ; \quad \phi_{ij} \neq 0\} \ . \tag{3}$$

Evidently, there is an one-to-one correspondence between the connectivity matrix Φ and $\{\mathcal{U}_j\}_{j=1}^N$.

The fusion rules that are the subject of this paper are algorithms that are implemented over a connected network topology and generate at the jth node a sequence of approximations $x_j[n]$ to the desired scalar computation $G(\mathbf{x})$. The class of rules that can be implemented over a given topology Φ are described by the following definition:

DEFINITION 1. *A set of rules, $\{F_j^{(n)}\}_{j=1}^N$, $n > 0$, will be referred to as an **admissible** distributed rule with respect to a given topology Φ, if*

$$x_j[n+1] = F_j^{(n)}(x_j,\, \{x_i[k];\quad k \leq n,\quad i \in \mathcal{U}_j(\Phi)\}) \ . \tag{4}$$

In this paper we focus on linear admissible rules, *viz*, rules of the form

$$\mathbf{x}[n] = \sum_{k \geq 1} W[n;k]\, \mathbf{x}[n-k], \qquad n > 0\,, \tag{5}$$

where $\mathbf{x}[n] \stackrel{\triangle}{=} \begin{bmatrix} x_1[n] & x_2[n] & \cdots & x_N[n] \end{bmatrix}^T$, and $W[n;k]$ is an $N \times N$ admissible matrix kernel. Admissibility of the rule (5) in the sense (4) places restrictions on the entries of $W[n;k]$. In particular, letting $W_{ij}[n;k] \stackrel{\triangle}{=} \{W[n;k]\}_{ij}$, we have $W_{ij}[n;k] = 0$, if $k \leq 0$, or if $\phi_{ij} = 0$.

A special and important subclass of these systems involves admissible LTI rules, *i.e.*, rules of the form (5) with

$$W[n;k] = W[1;k] \stackrel{\triangle}{=} W[k]\,, \tag{6}$$

and where the $N \times N$ matrix sequence $W[n]$ is admissible in the sense (4), *i.e.* $W_{ij}[n] = 0$, if $n \leq 0$, or if $\phi_{ij} = 0$.

We are interested in developing admissible rules which asymptotically compute a desired scalar function $G(\mathbf{x})$.

DEFINITION 2. *An admissible rule on a topology Φ is **asymptotically converging** (AC) to the desired scalar function $G(\mathbf{x})$, if the sequence $\mathbf{x}[n]$ satisfies*

$$\lim_{n\to\infty} ||\mathbf{x}[n] - \mathbf{1}G(\mathbf{x})|| = 0, \tag{7}$$

where $||\cdot||$ is the ordinary Euclidean norm, and $\mathbf{1}$ is $N \times 1$ vector of 1's.

Our objective in this paper is to design AC rules by exploiting only *locally available* information at each node. In selecting the rules, we will assume that the jth node has available $\mathcal{U}_j$ in (3), but *not* the full network topology, Φ. We will also consider the use of additional forms of macroscopic information about the network topology, including estimates of the average and squared-average number of connections in the network.

3. ASYMPTOTICALLY CONVERGING RULES FOR COMPUTING AVERAGES

In this section we develop AC rules for computing averages of the node data by only exploiting locally available information at the nodes. Without loss of generality we consider distributed computation of the function

$$G(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\mathbf{1}^T\mathbf{x}\ . \tag{8}$$

In particular, this type of rule and its linearly weighted generalizations can be used for computations that arise in a variety of problems involving signal estimation in noise, including problems of source detection classification and tracking [13]. We first develop and analyze a class of locally constructed first-order LTI rules for computation of (8). Then we extend our results to develop improved constructions of AC LTI rules.

3.1 Asymptotically Converging First-Order LTI Rules

In this section we develop methods for constructing admissible first-order LTI AC rules based on locally available information. We first present a set of sufficient conditions for asymptotic convergence, and then develop simple locally constructed algorithms that asymptotically converge to the desired $G(\mathbf{x})$ in (8).

First-order admissible LTI rules are special cases of (5) corresponding to $W[n;k] = W\,\delta[k-1]$, where W is an $N\times N$ admissible matrix, *i.e.*, satisfying $W_{ij} = 0$ for all $\phi_{ij} = 0$. In this case, (5) reduces to

$$\mathbf{x}[n] = W\,\mathbf{x}[n-1]\ . \tag{9}$$

We use the notation $W(\Phi)$ when it is deemed necessary to explicitly denote the dependence of W on Φ. In light of the target computation (8), we consider the initialization of the recursion (9) via

$$\mathbf{x}[n] = \mathbf{x}, \quad \text{for } n \leq 0\,, \tag{10}$$

and where $\mathbf{x}$ is given by (1). This initialization is admissible in the sense of (4), as it amounts to setting $x_i[n] = x_i$ for all $n \leq 0$ and all $1 \leq i \leq N$. The following set of conditions on $W(\Phi)$ guarantees asymptotic convergence to (8):

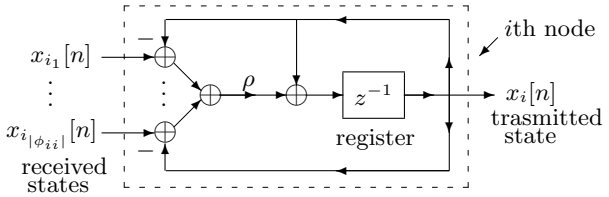

Figure 2: Block diagram for a first-order admissible rule at the ith node.

$$W(\Phi)\,\mathbf{1} \;=\; W(\Phi)^T\,\mathbf{1} \;=\; \mathbf{1} \tag{11a}$$

$$\lambda_1 \;=\; 1 \tag{11b}$$

$$|\lambda_i| \;<\; 1, \qquad \text{for } \; 2 \leq i \leq N\,, \tag{11c}$$

where $\{\lambda_i\}_{i=1}^N$ denote the eigenvalues of W in nonincreasing order.

For any arbitrary but fixed connected topology Φ, conditions (11a)–(11b) can be guaranteed by selecting the entries of W so that $W_{ij} = W_{ji}$ (with the need for $W_{ij} = W_{ji} = 0$ if $\phi_{ij} = 0$) and $W_{ii} = 1 - \sum_{j\neq i} W_{ij}$. Achieving condition (11c) is dictated by the choice of the nonzero off-diagonal entries of W.

The simple choice $W_{ij} = \rho\,\phi_{ij}$ for all $i \neq j$ (and $W_{ii} = 1-\rho\sum_{j\neq i}\phi_{ij}$ for all i) yields asymptotically converging rules for proper choice of the uniform diffusion (UD) parameter ρ. Fig. 2 shows a block diagram for an implementation of this rule at the ith node. This implementation form reveals strong connections between this class of rules and networks of coupled nonlinear oscillators used to explain global synchronization in biological species. In particular, similar to these global-synchronization models, these UD rules possess the properties of reciprocity and balancing. Reciprocity corresponds to each pair of connected nodes using the same fraction of each other's state in their computation (ρ if the nodes are connected and zero otherwise), while balancing corresponds to ensuring that the sum of all fractions used in adjusting the state of any particular node is zero.

Using (2), these UD rules can be expressed in the following convenient form

$$W(\Phi,\rho) = I + \rho\,\Phi\ . \tag{12}$$

Due to the structure of Φ, any rule of the form (12) satisfies conditions (11a)–(11b). In particular, due to (2), the eigendecomposition of Φ is of the form $\Phi = V\,M\,V^T$, which, when combined with (12) yields

$$W = V\Lambda V^T \tag{13}$$

where $\lambda_i = \{\Lambda\}_{ii} = 1 + \rho\mu_i$, and where $\{\mu_i\}_{i=1}^N$ are the eigenvalues of Φ (diagonal elements of M) in nonincreasing order. Since for connected topologies, (as it can be readily verified) $\mu_1 = 0$ and $\mu_i < 0$ for all $i > 1$, condition (11c) is satisfied if and only if $\rho \in (0, -\frac{2}{\mu_N})$, *i.e.*,

$$|\lambda_i| < 1, \;\; 2 \leq i \leq N \quad \Leftrightarrow \quad \rho \in (0,\, \rho_{\rm UB}) \tag{14}$$

where $\rho_{\rm UB} \triangleq -\frac{2}{\mu_N}$.

The choice of ρ in (12) determines the rate of convergence of the sequence $\mathbf{x}[\mathbf{n}]$ to the desired objective function (8).

Letting

$$\tilde{\mathbf{x}} = V^T \mathbf{x}\,, \tag{15}$$

where V is given by (13), the squared-error at step n incurred by the rule (12) is given in terms of $\tilde{x}_i = \{\tilde{\mathbf{x}}\}_i$ as follows

$$||\mathbf{x}[n] - \mathbf{1}\, G(\mathbf{x})||^2 = \sum_{i=2}^{N} \lambda_i^{2n} |\tilde{x}_i|^2 = \sum_{i=2}^{N} (1+\rho\,\mu_i)^{2n} |\tilde{x}_i|^2\,, \tag{16}$$

and is dominated for large n by the slowest decaying mode, $\max_{2\le i\le N} |\lambda_i|$. The choice of ρ that results in minimizing $\max_{2\le i\le N} |\lambda_i|$, yields $\lambda_2 = -\lambda_N$ and is given by

$$\rho_{\text{minmax}} = -\frac{2}{\mu_2 + \mu_N}\ . \tag{17}$$

In general, the choice of ρ_{minmax} requires knowledge of Φ (or, its eigenvalues) and does not lead to locally constructed rules.

Good selections of ρ can be made by exploiting macroscopic and/or locally available information about the network topology. One such choice corresponds to

$$\rho_{\max-\text{con}} = \frac{1}{\max_i |\phi_{ii}|}\ . \tag{18}$$

Indeed, as $|\phi_{ii}|$ denotes the number of connections of the ith node, $\max_i |\phi_{ii}|$ can be obtained via a finite-step (distributed) voting scheme for obtaining the maximum of the $|\phi_{ii}|$'s. Furthermore,

$$\max_i |\phi_{ii}| + 1 \le |\mu_N| \le 2 \max_i |\phi_{ii}|\,, \tag{19}$$

and where the upper bound on $|\mu_N|$ is readily established by using Gersgorin's theorem ([9], pp. 344–348). Consequently, we have $0 < \rho_{\max-\text{con}} \le \rho_{\text{UB}}$. Although, as we show in App. A, network topologies exist for which $\rho_{\max-\text{con}} = \rho_{\text{UB}}$, thereby leading to rules that do not satisfy (14), in general the choice (18) yields AC rules.[1] However, the choice $\rho_{\max-\text{con}}$ typically yields eigenvalues in (13) with larger magnitudes than can be achieved with the optimal choice ρ_{minmax} from (17).

3.2 Asymptotically Converging LTI Rules

First-order AC rules can be used to design more general LTI admissible rules that are asymptotically converging with improved convergence rates. In particular, given an admissible first order rule described by W satisfying (11), we consider LTI rules of the form

$$W[k] = W\, h[k] \tag{20}$$

where $h[k]$ is an arbitrary strictly-causal scalar sequence. These rules are admissible in the sense (4) provided W is admissible. Fig. 3 shows the local implementation of the rule at the ith node corresponding to (20) in terms of the causal filter $\tilde{H}(z) = z\,H(z)$, in the case that W is given by (12).

Proper choice of the strictly causal filter $h[k]$ in (20) with W in the form (12) can lead to AC rules with faster convergences rates than the associated first-order rules. In particular, provided

$$\sum_k h[k] = 1,$$

[1]We remark that we can readily construct AC rules for arbitrary connected topologies by adding to the denominator of the fraction on the right-hand side of (18) an $\epsilon > 0$.

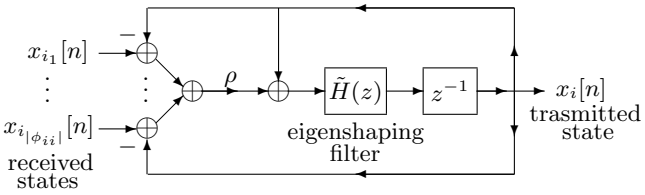

Figure 3: Block diagram for admissible rule at the ith node corresponding to (20) with W given by (12) and where $\tilde{H}(z) = z\,H(z)$ is a causal filter.

and assuming that W is of the form (12) and satisfies (14), the vector $\mathbf{x}[0] = \mathbf{1}$ is an eigenvector of the LTI rule described via (5)–(6), (20), and (12), initialized via (10), with 1 as the associated eigenvalue. Consequently, the objective is to select $h[k]$ so that the rest of the resulting eigenvalues are, in magnitude, as small as possible. To this end consider the sequence

$$\tilde{\mathbf{x}}[n] = V^T \mathbf{x}[n] \tag{21}$$

where V is given by (13). Premultiplying each side of (5) by V^T, and using (6), (20) and (21) in (5), we obtain a set of N decoupled scalar equations

$$\tilde{x}_i[n] = \lambda_i \sum h[k]\, \tilde{x}_i[n-k] + v_i[n]\ , \tag{22}$$

for $1 \le i \le N$ and $n > 0$, and where $v_i[n] \stackrel{\triangle}{=} \tilde{x}_i\, u[-n]$, and $\tilde{x}_i = \{\tilde{\mathbf{x}}\}_i$ from (15). Equivalently, as (22) suggests, we may view $\tilde{x}_i[n]$ for $n > 0$ as the response of an LTI system with transfer function $D_{\lambda_i}(z)$ to the input $v_i[n]$, where

$$D_\lambda(z) \stackrel{\triangle}{=} \frac{1}{1 - \lambda\, H(z)}\ .$$

Maximizing the rate of convergence of $|\tilde{x}_i[n]|^2$ to 0, is equivalent to minimizing the spectral radius of the denominator of $D_{\lambda_i}(z)$.

A simple yet effective class of filters yielding AC LTI rules is given by

$$H(z) = \frac{(1+c)\, z^{-1}}{1 + c\, z^{-2}}\ , \tag{23}$$

for some $0 \le c < 1$. These filters yield a spectral radius (magnitude of maximum-magnitude pole of $D_\lambda(z)$)

$$\beta(\lambda) = \begin{cases} \sqrt{c} & \text{if } |\lambda| \le \lambda_o \\ \frac{|\lambda|(1+c) + \sqrt{\lambda^2(1+c)^2 - 4c}}{2} & \text{if } \lambda_o \le |\lambda| \le 1 \end{cases},$$

where $\lambda_o = 2\sqrt{c}/(c+1)$. The reshaping of the eigenvalues $\{\lambda_i\}_{i=1}^N$ of W, induced by a filter of the form (23) is shown in Fig. 4. As the figure reveals, these filters increase the spectral radius for λ_i's for which $|\lambda_i| < \sqrt{c}$, at the benefit of decreasing the spectral radius for large magnitude λ_i's, *i.e.*, for $|\lambda_i| > \sqrt{c}$.

Fig. 5 shows the eigenvalue-magnitude distribution of four different rules applied on the network topology in Fig. 1. The top-left and top-right graphs in Fig. 5 show the eigenvalue distribution for the first-order rules described via (9) and (12) initialized via (10), with $\rho = \rho_{\text{minmax}}$ and $\rho = \rho_{\max-\text{con}}$, respectively. The bottom-left and bottom-right graphs on Fig. 5 show the eigenvalue distribution of the LTI rules described via (5)–(6), (20), (12), and (23) initialized

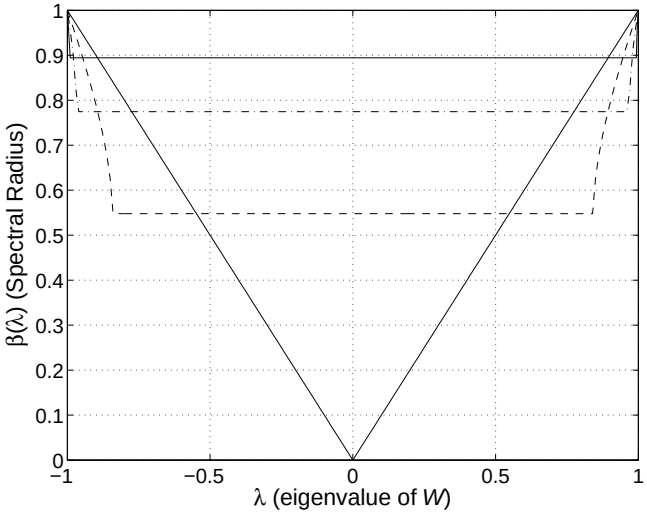

Figure 4: Eigenvalue shaping by the filter (23) with $c = 0.1$ (dashed), $c = 0.3$ (dash-dot), and $c = 0.6$ (solid). For comparison purposes, the lines $|\lambda|$ (solid), corresponding to the case $c = 0$ (*i.e.*, no filtering) are also shown.

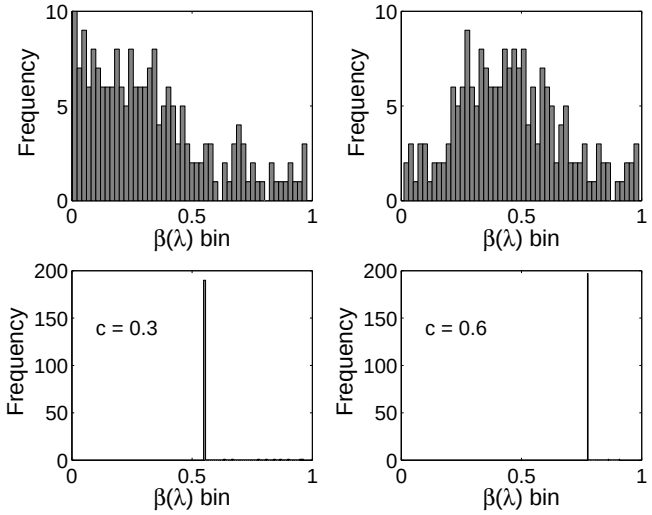

Figure 5: Eigenvalue-magnitude distribution of four LTI rules applied on the topology depicted in Fig. 1.

via (10) with $\rho = \rho_{\max-\mathrm{con}}$ for two values of the filter parameter c. As the figure demonstrates, filtering at each node via $H(z)$ reduces the large-magnitude modes in the system at the expense of low-magnitude modes. In light of (16), these rules would outperform, in general, the associated first-order rules.

Finally, due to the initialization (10), a rule with $H(z)$ from (23) can also be viewed as a time-varying rule form (5) with various choices for the kernel $W[n;k] = h[n;k]\,W$. Although all these rules are equivalent in terms of the output sequence $\mathbf{x}[n]$ that they yield, the choice with kernel

$$W[n;k] = \begin{cases} W\,\delta[k-1] & \text{if } n=1 \\ (1+c)\,W\,\delta[k-1] - c\,W\,\delta[k-2] & \text{if } n=2 \\ (1+c)\,W\,\delta[k-1] - c\,I\,\delta[k-2] & \text{if } n>2 \end{cases} \tag{24}$$

is preferable for implementation, due to its finite-memory computation requirements.

4. DISTRIBUTED SIGNAL ESTIMATION IN GAUSSIAN NOISE

In this section we characterize the performance of the admissible AC rules presented in Sec. 3 in the context of a driving example involving signal estimation for noisy signal observations at the network nodes. In particular, we consider estimation of an unknown parameter A based on observation of a vector $\mathbf{x} \sim \mathcal{N}(\mathbf{1}A, \sigma^2 I)$, where I is the $N \times N$ identity matrix, and $\mathbf{x}$ is given by (1). The minimum variance unbiased estimator (MVUE) of A based on $\mathbf{x}$ is given by

$$\hat{A}(\mathbf{x}) = \frac{1}{N}\mathbf{1}^T\mathbf{x}\ . \tag{25}$$

and yields the following mean-square error (MSE)

$$\sigma^2_{\hat{A}} \triangleq \mathrm{var}\,\hat{A} = \frac{\sigma^2}{N}\ . \tag{26}$$

As (25) corresponds to computation of (8), the AC rules of Sec. 3 of the form (5), initialized via (10), can be used to obtain a sequence of unbiased estimates $\tilde{A}_i[n]$ at the ith node, which asymptotically converges to $\hat{A}(\mathbf{x})$ in (25). To this end, we consider the nth step $N \times 1$ vector estimate sequence

$$\tilde{\mathbf{A}}[n] = \begin{bmatrix} \tilde{A}_1[n] & \tilde{A}_2[n] & \cdots & \tilde{A}_N[n] \end{bmatrix}^T\ , \tag{27}$$

where $\tilde{A}_i[n] = x_i[n]$ from (5). The total MSE associated with $\tilde{\mathbf{A}}[n]$ is given by $\mathrm{tr}\Lambda_{\tilde{\mathbf{A}}}[n]$ (the trace of $\Lambda_{\tilde{\mathbf{A}}}[n]$), where

$$\Lambda_{\tilde{\mathbf{A}}}[n] \triangleq E\left[\left(\tilde{\mathbf{A}}[n] - \mathbf{1}\,A\right)\left(\tilde{\mathbf{A}}[n] - \mathbf{1}\,A\right)^T\right]\ . \tag{28}$$

The quality of finite-delay approximations to (25) provided by a distributed rule (5) can be characterized by the relative MSE induced by the distributed rule after n iterations, *i.e.*,

$$\mathrm{RMSE}\left(\tilde{\mathbf{A}}[n]\right) \triangleq \frac{\frac{1}{N}\,\mathrm{tr}\Lambda_{\tilde{\mathbf{A}}}[n] - \sigma^2_{\hat{A}}[n]}{\sigma^2_{\hat{A}}[n]} = \sigma^{-2}\,\mathrm{tr}\Lambda_{\tilde{\mathbf{A}}}[n] - 1\ . \tag{29}$$

The relative MSE denotes the relative additional MSE incurred by the distributed algorithm, on average, as compared to the desired rule (25).

In the case of first-order admissible AC rules based on (12), the form of the relative MSE can be readily determined. In particular, by using (16) we obtain

$$\mathrm{RMSE}\left(\tilde{\mathbf{A}}[n]\right) = \sum_{i=2}^{N}(1+\rho\,\mu_i)^{2n}\ . \tag{30}$$

As (30) reveals, the relative MSE performance depends on the network topology, through the eigenvalues of Φ, and the choice of the diffusion parameter ρ. The value of ρ that minimizes the relative MSE for a given Φ after n iterations, *i.e.*,

$$\rho_n = \arg\min_{\rho>0} \sum_{i=2}^{N}(1+\rho\,\mu_i)^{2n} \tag{31}$$

is a function of n. In the limit $n \to \infty$, we have

$$\rho_\infty \triangleq \lim_{n\to\infty} \rho_n = \rho_{\text{minmax}} \ .$$

Attaining ρ_n in (31) for some fixed n requires, in general, knowledge of the network topology matrix Φ. For $n = 1$, the minimization in (31) yields

$$\rho_1 = -\frac{\sum_{i=1}^N \mu_i}{\sum_{i=1}^N \mu_i^2} \ , \tag{32}$$

which can be computed via macroscopic network information. In particular, due to the structure of Φ,

$$\frac{1}{N}\sum_{i=1}^N \mu_i = \frac{1}{N}\sum_{i=1}^N \phi_{ii}, \qquad \frac{1}{N}\sum_{i=1}^N \mu_i^2 = \frac{1}{N}\sum_{i=1}^N \phi_{ii}^2 - \frac{1}{N}\sum_{i=1}^N \phi_{ii} \ ,$$

i.e., the numerator and the denominator of (32) can be obtained via the average and squared-average number of connections in the network. Thus, (32) can be rewritten as

$$\rho_1 = \frac{\sum_{i=1}^N |\phi_{ii}|}{\sum_{i=1}^N |\phi_{ii}| + \sum_{i=1}^N |\phi_{ii}|^2}. \tag{33}$$

Although, as we show in App. B, network topologies can be constructed for which ρ_1 does not yield an AC rule, for large networks this choice yields, in general, AC rules with eigenvalue distribution superior to the one based on (18).

The relative MSE incurred by LTI rules described by (5)–(6), (20), (12), and (23), initialized via (10) in computing the MVUE (25) can also be obtained in closed form. In particular, by applying the energy-preserving transformation $\tilde{x}[n] = V^T \mathbf{x}[n]$ and using (24), we obtain

$$\text{RMSE}\left(\tilde{\mathbf{A}}[n]\right) = \sum_{i=2}^N \lambda_i^2 \ (a_{i1}\,\xi_{i1}^n + a_{i2}\,\xi_{i2}^n)^2 \ , \tag{34}$$

where

$$\xi_{i1} = \frac{(1+c)\,\lambda_i + \sqrt{(1+c)^2\lambda_i^2 - 4\,c}}{2} \ ,$$

$$\xi_{i2} = \frac{(1+c)\,\lambda_i - \sqrt{(1+c)^2\lambda_i^2 - 4\,c}}{2} \ ,$$

$$a_{i1} = u_i\,(\xi_{i1} - \xi_{i2})^{-2}(\xi_{i1}\gamma_i - \xi_{i1}\xi_{i2}) \ ,$$

$$a_{i2} = -u_i\,(\xi_{i1} - \xi_{i2})^{-2}(\xi_{i2}\gamma_i - \xi_{i1}\xi_{i2}) \ ,$$

and $\gamma_i \triangleq (1+c)\lambda_i - c$, $u_i \triangleq \sqrt{|\xi_{i1}|^2+1}\sqrt{|\xi_{i2}|^2+1}$, and $\lambda_i = 1 + \rho\mu_i$.

5. SIMULATIONS

In this section we present a simulation-based performance evaluation of the rules of Sec. 3. All our simulations involve networks with N nodes uniformly distributed on a disk of unit radius, where the probability that any two nodes i and j are connected, is a function of their distance, d_{ij}. In particular, we employ a two-parameter model according to which nodes i and j are connected with probability

$$\Pr\left[\phi_{ij} = 1\right] \ = \ 2^{-\left(\frac{d_{ij}}{d_{\text{nom}}}\right)^m} \ , \tag{35}$$

for some $m \geq 1$ and $d_{\text{nom}} > 0$. Although the above choice of a connectivity model is somewhat arbitrary, it has various desirable properties. First, $\Pr\left[\phi_{ij} = 1\right]$ is a decreasing function of d_{ij}, with $\Pr\left[\phi_{ij} = 1\right] = 1/2$ at $d_{ij} = d_{\text{nom}}$. As a result, according to model (35), closely located nodes are more likely to be connected. Also, as m increases, $\Pr\left[\phi_{ij} = 1\right]$ increases toward 1 for any $d_{ij} < d_{\text{nom}}$, and decreases toward 0 for any $d_{ij} > d_{\text{nom}}$. Hence, in the limit $m \to \infty$, all node pairs with distance less than d_{nom} are connected, while all node pairs with distance greater than d_{nom} are not. We remark that the general trends observed in the simulations are not a feature of the choice of the connectivity model (35) and are typical of other models.

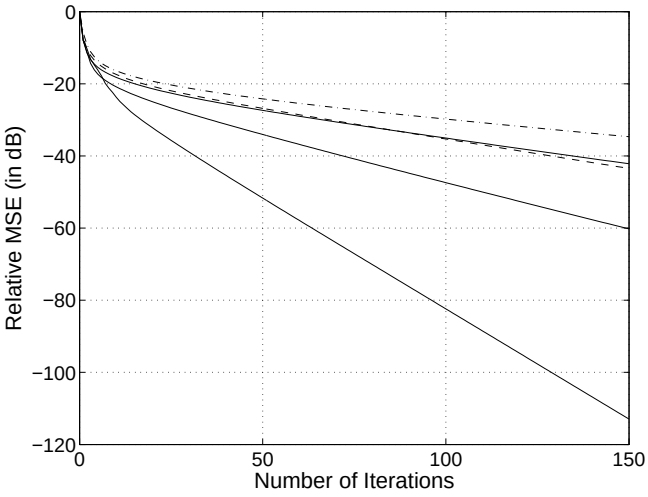

Figure 6: Relative MSE incurred by various AC LTI rules for computing (25), applied on the network in Fig. 1.

Fig. 6 shows the relative MSE in the Gaussian estimation problem of Sec. 4, incurred by various LTI AC rules for the 200-node network in Fig. 1. The network was constructed according to model (35) with $d_{\text{nom}} = 1/4$ and $m = 6$. The dashed and dash-dot curves show the relative MSE performance of the first order rules (9)–(10) and (12), with $\rho = \rho_{\text{minmax}}$ and $\rho_{\text{max-con}}$, respectively. The solid curves depict the relative MSE performance of the LTI rule described by (5)–(6), (20), (12), and (23) initialized via (10) with $\rho = \rho_1$ for different values of the parameter c. Specifically, the successively steeper solid curves show the relative MSE performance of the algorithm for $c = 0$ (no filtering), $c = 0.3$, and $c = 0.6$, respectively. As the figure demonstrates, the first-order rules with $\rho = \rho_1$ and $\rho = \rho_{\text{minmax}}$ have similar relative MSE performance characteristics on this network. Furthermore, filtering can significantly reduce the relative MSE achieved after a given number of iterations, and can thus reduce the number of iterations needed to achieve a desired relative MSE level. Alternatively, the LTI rule with $c = 0.6$, can reach any given target RMSE level, in a fraction of the number of iterations needed by the associated first-order rule ($c = 0$) to reach the same target RMSE.

The improvement in the convergence rate due to the presence of a filter $H(z)$ of the form (23) can be quantified by means of the gain provided by $H(z)$ in terms of the number of iterations required to achieve a given quality of approximation. To this end, we define the iteration-gain factor as

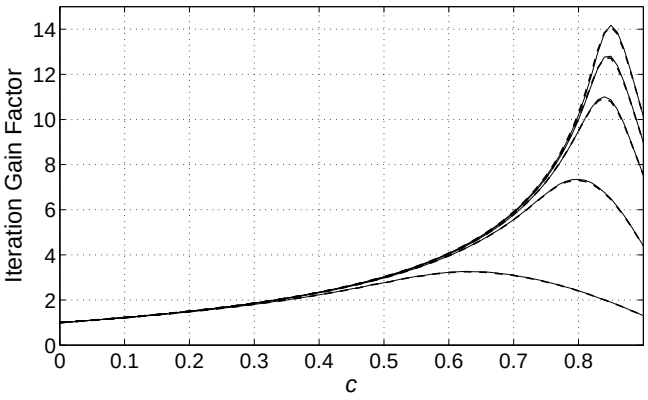

Figure 7: Iteration-gain factors provided by $H(z)$ in (23) in terms of the iterations needed to achieve a target relative MSE in networks with $N = 400$ nodes, for target relative MSEs -20, -30 -40, -50, and -60 dB.

the ratio of the number of iterations needed by a first-order rule over the number of iterations needed by the associated LTI rule based on $H(z)$ to reach a target level of relative MSE, *i.e.*,

$$\gamma(D, c, \Phi, \rho) \triangleq \frac{n_0(D, \Phi, \rho)}{n_c(D, \Phi, \rho)}\,, \tag{36}$$

where $n_c(D, \Phi, \rho)$ is the number of iterations needed by the LTI rule described by (5)–(6), (20), (12), and (23) initialized via (10) with a diffusion parameter ρ and a filter parameter c to reach a target RMSE of D dB over a topology Φ, while $n_0(D, \Phi, \rho)$ is the number of iterations required by the associated first-order rule (LTI rule with $c = 0$, or, equivalently, the rule described via (9), with W given by (12) and initialized via (10)) to reach the same target RMSE level over the same topology.

Fig. 7 shows iteration-gain factor estimates obtained via $L = 100$ network realizations, $\{\Phi_i\}_{i=1}^{L}$, for various levels of target relative MSE D (in dB). All networks have $N = 400$ nodes and were generated according to model (35) with $d_{\rm nom} = 0.15$ and $m = 6$. All the curves in the figure correspond to AC rules that employ $\rho = \rho_{\max-\rm con}$ from (18). The successively higher-gain solid curves in the figure depict the sample-mean iteration-gain factor, *i.e.*,

$$\bar{\gamma}(D, c, \rho) = \frac{1}{L}\sum_{i=1}^{L} \gamma(D, c, \Phi_i, \rho) \tag{37}$$

as a function of the filtering parameter c, for target relative MSEs -20, -30, -40, -50 and -60 dB, respectively. The associated dashed curves correspond to the iteration-gain factor estimates obtained by taking the ratio of the average number of iterations needed by the first-order rule to achieve the target RMSE level, over the average of the number of iterations needed the associated LTI rule; *viz*,

$$\tilde{\gamma}(D, c, \rho) = \frac{\frac{1}{L}\sum_{i=1}^{L} n_0(D, \Phi_i, \rho)}{\frac{1}{L}\sum_{i=1}^{L} n_c(D, \Phi_i, \rho)}\,. \tag{38}$$

As the figure demonstrates, use of the filter (23) can provide significant improvements in the convergence rates of the distributed algorithm. Furthermore, the choice of the filter parameter c in (23) that optimizes the convergence rates is a function of the target relative MSE. Finally, both (37) and (38) yield effectively equivalent estimates of the iteration-gain factors achieved, revealing that these gains are consistent over different network realizations.

6. ASYMPTOTICALLY CONVERGING RULES FOR OTHER COMPUTATIONS

The distributed fusion rules of Sec. 3 can be readily generalized to compute a broader class of functions. First, they can be employed to compute averages of functions of the data of the form

$$G(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) \tag{39}$$

for arbitrary $f_i(\cdot)$. In particular, given an AC rule of the form (5) initialized with (10) for computing $G(\mathbf{x})$ in (8), we may obtain an AC rule for computing a $G(\mathbf{x})$ of the form (39) by replacing the initialization step (10) with

$$\mathbf{x}[n] = \begin{bmatrix} f_1(x_1) & f_2(x_2) & \cdots & f_N(x_N) \end{bmatrix}^T \qquad n \leq 0\,.$$

The quality of approximations provided by these algorithms subject to finite-delay computations can be evaluated, in principle, using techniques similar to those in Sec. 4. In general, the associated approximation quality is dictated by $f_i(\cdot)$ and the prior on $\mathbf{x}$, and is, hence, case and context specific. It is interesting to remark that computations of the form (39) can arise in a number of detection and estimation problems, where, often, the functions $f_i(\cdot)$ in the global computation (39) are not locally available, but rather depend on the quality of the measurements throughout the network. One such example arises in the context of estimation of a signal parameter A based on observation of $\mathbf{x} \sim \mathcal{N}(A\mathbf{b}, \sigma^2 I)$, whereby node i only knows b_i but not the rest of the elements of $\mathbf{b}$. The MVUE of A in that case takes the form

$$\hat{A}(\mathbf{x}) = \frac{\mathbf{b}^T\mathbf{x}}{||\mathbf{b}||^2}\,, \tag{40}$$

and can be recast in the form (39), where $f_i(x) = \alpha_i\, x$, with

$$\alpha_i = \frac{b_i}{\sigma_b^2(\mathbf{b})}\,,$$

and where

$$\sigma_b^2 = \sigma_b^2(\mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N} b_i^2\,. \tag{41}$$

Given an AC rule (5) initialized with (10) for computing (8), we may obtain an AC rule for computing σ_b^2 in (41) by replacing the initialization step (10) with

$$\mathbf{x}[n] = \begin{bmatrix} b_1^2 & b_2^2 & \cdots & b_N^2 \end{bmatrix}^T \qquad n \leq 0\,.$$

Assuming sufficient convergence of the algorithm to $\sigma_b^2(\mathbf{b})$, the same AC rule can be used to compute (40), via the initialization

$$x_i[n] = \frac{b_i}{\hat{\sigma}_b^{(i)}}\, x_i \qquad n \leq 0, \quad 1 \leq i \leq N\,,$$

where $\hat{\sigma}_b^{(i)}$ denotes the approximation of σ_b^2 at node i via the distributed computation of σ_b^2.

The class of functions $G(\cdot)$ that can be computed via the distributed algorithms presented in Secs. 2–3 also includes functions that can be expressed as compositions of an invertible transformation with $G(\mathbf{x})$ in (39). One such example involves computation of the geometric mean

$$G_{\rm gm}(\mathbf{x}) = \left(\prod_{i=1}^{N} x_i\right)^{\frac{1}{N}} = \exp\left\{\frac{1}{N}\sum_{i=1}^{N} \ln x_i\right\}. \qquad (42)$$

Given an AC rule (5) that computes (8), we can obtain $G_{\rm gm}(\mathbf{x})$ by first applying the AC rule with initialization $x_i[n] = \ln x_i$ for $n \leq 0$, and then obtaining the approximating sequence to $G_{\rm gm}(\mathbf{x})$ at the ith node as $\exp\{x_i[n]\}$.

7. CONCLUDING REMARKS

We developed distributed algorithms for performing a class of global computations in ad-hoc networks. These algorithms provide sequences of approximations to the desired global computation at each node and are constructed using only locally available information about the network topology. As we have shown, these distributed algorithms can be locally designed, so as to optimize the convergence rates of the approximating sequences at each node to the desired computation.

Due to their locally constructed and optimized nature, these algorithms can accommodate changing topologies, of the type that are encountered in networks of wireless, potentially mobile, nodes, as they only require that nodes know and incorporate local changes in the network topology. Furthermore, these algorithms are inherently scalable and fault tolerant.

The distributed algorithms we presented allow asymptotic computation of weighted global averages of the node data. More generally, the approach we presented can be extended to compute a broad range of global computations arising in data fusion applications, by decomposing them into sets of weighted average computations. One such example involves distributed implementation of source localization algorithms based on sensor measurements that provide information about the relative range of the source from each sensing node [13].

8. REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Commun. Mag.*, pages 102–114, Aug. 2002.

[2] P. Bauer, M. Sichitiu, R. Istepanian, and K. Premaratne. The mobile patient: Wireless distributed sensor networks for patient monitoring and care. In *Proc. 2000 IEEE EMBS Int. Conf. on Info. Tech. App. in Biomedicine*, pages 17–21, 2000.

[3] R. Chellappa, Q. Zheng, P. Burlina, C. Shekhar, and K. Eom. On the positioning of multisensor imagery for exploitation and target recognition. *Proc. IEEE*, 85(1):120–138, Jan. 1997.

[4] B. Ermentrout. An adaptive model for synchrony in the firefly *pteroptyx malaccae*. *J. Math. Biology*, 29(6):571–585, June 1991.

[5] K. Friedman, W. Pichel, and X. Li. Mesoscale oceanic and atmospheric feature detection through fusion of RADARSAT SAR with GOES/Imager data. In *Int. Symp. Geosci. Remote Sensing*, pages 907–909, July 1998.

[6] S. Grime, H. Durrant-Whyte, and P. Ho. Communication in decentralized data-fusion systems. In *Proc. Amer. Commun. Conf.*, pages 3299–3303, 1992.

[7] M. Honikel. Improvement of InSAR DEM accuracy using data and sensor fusion. In *Int. Symp. Geosci. Remote Sensing*, pages 2348–2350, July 1998.

[8] F. Hoppensteadt and E. Izhikevich. Pattern recognition via synchronization in phase-locked loop neural networks. *IEEE Trans. Neural Networks*, 11(3):734–738, May 2000.

[9] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *MobiCom*, pages 56–67, Boston, MA, 2000.

[11] M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proc. IEEE*, 85(1):108–119, Jan. 1997.

[12] C. Nelson and D. Fitzgerald. Sensor fusion for intelligent alarm analysis. *IEEE Aero. and Elec. Sys. Mag.*, 12(9):18–24, Sept. 1997.

[13] T. Pham, D. S. Scherber, and H. C. Papadopoulos. Distributed source localization algorithms for acoustic ad-hoc sensor networks. Submitted to 2004 Sensor Array and Multichannel Signal Processing Workshop.

[14] E. Preston, T. Bergman, R. Gorenflo, D. Herman, E. Kopala, T. Kuzma, L. Lazofson, and R. Orkis. Development of a field-portable imaging system for scene classification and multispectral data fusion algorithms. *IEEE Aero. Elec. Sys. Mag*, 9(9):13–19, Sept. 1995.

[15] H. Qi, P. Kuruganti, and Y. Xu. The development of localized algorithms in wireless sensor networks. *Sensors*, pages 286–293, July 2002.

[16] B. Rao, H. Durrant-Whyte, and J. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *Int. J. Robot. Res.*, 12(1):20–44, Feb. 1993.

[17] K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self organization of a wireless sensor network. *IEEE Personal Commun. Mag.*, pages 16–27, Oct. 2000.

[18] R. Viswanathan and P. Varshney. Distributed detection with multiple sensors: Part I – Fundamentals. *Proc. IEEE*, pages 54–63, Jan. 1997.

[19] M. Yeddanapudi, Y. Bar-Shallom, and K. Pattipati. IMM estimation for multitarget-multisensor air traffic surveillance. *Proc. IEEE*, 85(1):80–94, Jan. 1997.

[20] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration. *IEEE Signal Processing Mag.*, pages 61–72, Mar. 2002.

APPENDIX

A. CONVERGENCE OF FIRST-ORDER LTI RULES WITH $\rho = \rho_{\max-\text{con}}$

In this section we present network topologies for which first-order rules with $\rho = \rho_{\max-\text{con}}$ in (18) do not yield asymptotically converging rules. These N-node topologies are described as follows:

- each node is connected to exactly two other nodes to form a ring topology;
- N is even.

Each of these topologies yields a topology matrix, Φ, with $|\phi_{ii}| = 2$ for all i. Consequently, $\max_i |\phi_{ii}| = 2$ which, when substituted in (18), yields

$$\rho_{\max-\text{con}} = \frac{1}{2} \ .$$

As it can be readily verified, any network topology matrix Φ corresponding to such networks with N even, has a maximum (in magnitude) eigenvalue $\mu_N = -4$, resulting in

$$\rho_{\text{UB}} = -\frac{2}{\mu_N} = \frac{1}{2} \, ,$$

thereby yielding $\rho_{\max-\text{con}} = \rho_{\text{UB}}$, *i.e.* a $\rho_{\max-\text{con}}$ value outside the ρ range in (14) for asymptotic convergence.

B. CONVERGENCE OF FIRST-ORDER LTI RULES WITH $\rho = \rho_1$

In this section we present network topologies for which first-order rules with $\rho = \rho_1$ in (32) do not yield asymptotically converging rules. These N-node topologies are described as follows:

- $N-1$ of the nodes form a ring topology;
- the Nth node is connected to all $N-1$ nodes.

As it can be readily verified, each of these topologies yields a topology matrix, Φ, with maximum (in magnitude) eigenvalue $\mu_N = -N$, and thus $\rho_{\text{UB}} = \frac{2}{N}$. The associated ρ_1 from (33) in this case is given by

$$\rho_1 = \frac{(N-1)+3\,(N-1)}{(N-1)^2+9\,(N-1)+(N-1)+3\,(N-1)} = \frac{4}{N+12} \ .$$

For $N > 12$ these network topologies yield $\rho_1 > \rho_{\text{UB}}$, *i.e.* a ρ_1 value outside the ρ range in (14) for asymptotic convergence.