

# Distributed Averaging in Sensor Networks Based on Broadcast Gossip Algorithms

Mauro Franceschelli, Alessandro Giua and Carla Seatzu.

## Abstract

In this paper we propose a new decentralized algorithm to solve the consensus on the average problem on sensor networks through a gossip algorithm based on broadcasts. We directly extend previous results by not requiring that the digraph representing the network topology is balanced. Our algorithm is an improvement respect to known gossip algorithms based on broadcasts in that the average of the initial state is preserved after each broadcast. The nodes are assumed to know their out-degree anytime they transmit information.

## Index Terms

Consensus, sensor networks, distributed averaging, gossip algorithms.

Published as:

Mauro Franceschelli, Alessandro Giua and Carla Seatzu, "Distributed Averaging in Sensor Networks Based on Broadcast Gossip Algorithms" IEEE Sensors Journal, Special Issue on Cognitive Sensor Networks, 2010.

This work has been partially supported by the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFOS-ICT-224498).

M. Franceschelli, A. Giua and C. Seatzu are with the Dept. of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy. Email: {mauro.franceschelli,giua,seatzu}@diee.unica.it.

## I. INTRODUCTION

In recent years a great effort has been directed to the study of the consensus problem — i.e. the problem of making the scalar states of a set of agents converge to the same value under local communication constraints [1], [2], [3], [4], [5], [6] — and of its many applications. One of such applications, namely wireless sensor networks and in general peer-to-peer networks, is now the focus of a huge amount of research in many disciplines of information technology. The reason why the distributed average problem has received great attention is that it allows to achieve tasks with a minimum overhead of communication since it requires only local information exchange between nodes directly connected, i.e. no routing is needed and so no congestion due to network traffic is generated. One of the networks in which this is desirable is the internet in which the availability of information on the average of local quantities generated by users behavior is of great relevance for statistical analysis, marketing, security and so on. If such objectives can be achieved without unnecessarily overloading network nodes and user bandwidth the relevance of such algorithms becomes clear.

A different kind of networks are wireless embedded sensor networks, intended to be composed of a huge number of cheap wireless sensors scattered around a target, be it a city, a forest, a war field or a polluted area. By definition if a wireless sensor is to be cheap it has to consume very little power for achieving its task and to this end the ability to retrieve the average of the measurements with only local packet exchange is of great relevance.

Many previous works on the consensus problem and gossip algorithms [2], [4], [5], [6], [7], [8], [9], [10], [11] are based on bidirectional communications and so represent the network through an undirected graph, possibly with a switching topology. In [8] a study of convergence times of gossip algorithms based on pairwise random communications is presented for different network topologies. In [10] consensus on the average in presence of intermittent links and noise is addressed. In [11] the problem of designing the topology to maximize the rate of convergence of average consensus is addressed taking into account communication costs and constraints.

The requirement of bidirectional communications requires synchronization between transmitter and receiver and some overhead required by the communication protocols like acknowledgments. Furthermore even if a set of nodes can communicate between each other, communications are inherently sequential and pairwise if they are not done in the form of broadcasts. An attempt

to use broadcasts in the distributed average problem has been made with gossip algorithms, the tradeoff of this approach is that agreement is only reached in the form of a random variable whose expectation corresponds to the average of the initial measurements and whose shape is deeply affected by the sequence in which the nodes perform broadcasts.

A different approach to this issue is the use of distributed Kalman filtering based on consensus [12], [13]. A couple of years ago this problem was solved by adapting the optimal Kalman gain of such filter with respect to the outflow of each node [14] to achieve consensus on the average on arbitrary strongly connected digraphs. The proposed technique was time-variant and proposed as a decentralized iterative algorithm with synchronized updates. In this paper, that is a journal version of [15], we propose an alternative approach based on gossip.

In [2] the study of consensus on digraphs was motivated by reduction in communication costs. Unfortunately the conclusion of the authors was that consensus on the average is achievable only for balanced digraphs, i.e. graphs in which the in-degree and out-degree of each node are the same.

Starting from this, we developed a new algorithm, with the same feature of Laplacian-based consensus, that can achieve the same objective for the wider class of arbitrary digraphs. This generalizes the consensus problem and allows a consistent reduction of complexity since it allows the use of only broadcasting as communication mean.

Furthermore wireless sensor networks are usually required to perform tasks more complex than just computing the average of some quantity. We argue that an algorithm that allows consensus on directed graphs can actually be implemented as simple and small "overhead" on normal communication between the sensors. For instance with the ZigBee protocol for wireless networks we have packets with a maximum payload of around 104 bytes, which it is clearly much more than what is required to just send a scalar integer value of 16 bits. We argue that such consensus protocol could have a more meaningful and real application if thought as network overhead for distributed estimation purposes that does not actually "increase" the load in the network. Since no specific acknowledge or response is required, no dedicated communication is required and only the usual communication due to data transfer between nodes for other purposes is needed. With the previous assumption while the nodes use only mono-directional communications, they always know their out-degree.

Finally, the proposed algorithm poses new theoretical questions on stability of gossip al-

gorithms since it is an instance of gossip algorithm in which local interactions are based on asymmetric, non-contractive matrices with possibly both positive and negative elements taken from a set all the products of which converge [16].

Note that, even if a formal proof of convergence of the algorithm we propose is missing, a series of simulations are presented to illustrate the effectiveness of the approach. In particular, different network topologies have been considered that are scalable in the number of nodes, and for such topologies the dependence of the convergence times upon the number of nodes is shown. Finally, the paper presents a series of simulations to compare the proposed algorithm with other gossip algorithms known in the literature.

The structure of the paper is as follows. In Section II some background on graph theory and the notation used in the paper is introduced. In Section III the main contribution of this paper is presented, namely an algorithm to solve the consensus on the average problem using asynchronous broadcasts. In Section IV the convergence properties of the proposed algorithm are discussed. In Section V the results of several numerical simulations are presented. In Section VI the concluding remarks are given.

## II. PROBLEM STATEMENT

We model the network of agents as a directed graph  $\mathcal{G}(t) = \{V, E(t)\}$ , with  $V = \{1, \dots, n\}$  the set of nodes (or vertices) that represent the agents,  $E(t) \subseteq \{V \times V\}$  the time varying edge set that encodes the network topology,  $(i, j) \in E(t)$  if and only if agent  $i$  may receive information from agent  $j$  at time  $t$ . In the following directed edges from  $j$  to  $i$  are considered to have their "tail" in  $j$  and the "head" in  $i$ .

The graph can be encoded through its  $n \times n$  *adjacency matrix*

$$A(t) = \{a_{i,j}(t)\} \text{ with } a_{i,j}(t) = \begin{cases} 1 & \text{if } (i, j) \in E(t); \\ 0 & \text{otherwise.} \end{cases}$$

The in-degree of a node corresponds to the number of "heads" incident in such node while the out-degree is the number of "tails" incident on it.

We define the two  $n \times n$  matrices

$$\Delta_{in}(t) = \text{diag}(\delta_{in,1}(t), \dots, \delta_{in,n}(t))$$

and

$$\Delta_{out}(t) = \text{diag}(\delta_{out,1}(t), \dots, \delta_{out,n}(t))$$

where  $\delta_{in,i}$  and  $\delta_{out,i}$ , for  $i = 1, \dots, n$ , are respectively the in-degree and out-degree of node  $i$ .

The *Laplacian* of a time-varying digraph is defined as

$$\mathcal{L}(t) = \Delta_{in}(t) - A(t). \quad (1)$$

It is a positive semi-definite matrix and weak diagonally row dominant. Defining  $\mathbf{0}$  and  $\mathbf{1}$  column vectors whose  $n$  elements are all, respectively, zeros and ones, we have that  $\mathcal{L}(t)\mathbf{1} = \mathbf{0}$  by construction.

To each node  $i$  for  $i = 1, \dots, n$  is associated a scalar  $x_i(t)$  with an arbitrary initial value  $x_i(0) = x_{i0}$ .

Furthermore we define the set of neighbors of node  $i$  as  $\mathcal{N}_i(t) = \{j : (j, i) \in E\}$  and with  $|\mathcal{N}_i(t)|$  its cardinality. We point out that since the graph is directed, node  $i$  may be a neighbor of node  $j$  while node  $j$  is not a neighbor of node  $i$ .

Note that the underlying topology of the network is *deterministic* and specifies the edges that *may* be selected by the gossip algorithm at any time. What is random is the selection of the node that performs a broadcast, involving only the edges connected to it. So at any time instant the interaction topology is a set of neighbors of the broadcaster node which is taken at random.

Our objective is to find a decentralized control law that satisfies the network topology constraints given by  $\mathcal{G}(t)$  and achieves consensus on the average on the initial states.

### III. CONSENSUS ON THE AVERAGE ON ARBITRARY DIGRAPHS

In our approach we associate to each node  $i$  for  $i = 1, \dots, n$ , in addition to  $x_i(t)$  on which value a consensus on the average is sought, a companion variable  $z_i(t)$  with initial value  $z_i(0) = 0$ . In the following we study a gossip algorithm based on mono-directional communications. Each node at each instant of time is then either transmitting information, receiving information or in an idle state.

**RULE 1, Transmitter state update, node  $i$**

$$\begin{cases} x_i(t+1) = x_i(t), \\ z_i(t+1) = 0. \end{cases} \quad (2)$$

**RULE 2, Receiver state update, node  $j \in \mathcal{N}_i(t)$**

$$\begin{cases} x_j(t+1) = \frac{x_j(t)+x_i(t)}{2} + 0.5z_j(t) + \frac{z_i(t)}{2\delta_{out,i}(t)}, \\ z_j(t+1) = \frac{x_j(t)-x_i(t)}{2} + 0.5z_j(t) + \frac{z_i(t)}{2\delta_{out,i}(t)}. \end{cases} \quad (3)$$

**RULE 3, Idle nodes,  $k \neq i, k \notin \mathcal{N}_i(t)$**

$$\begin{cases} x_k(t+1) = x_k(t), \\ z_k(t+1) = z_k(t). \end{cases} \quad (4)$$

*Algorithm 1 (Extended Gossip based on Broadcasts (EGB)):*

- 1) Let  $t=0$ , let  $\mathbf{x}(0) = \mathbf{x}_0$  and  $\mathbf{z}(0) = \mathbf{0}$ .
- 2) A node  $i$  at random executes **RULE 1**
- 3) Each node  $j \in \mathcal{N}_i(t)$  that listens to the broadcast applies **RULE 2**.
- 4) All the other nodes  $k \neq i, k \notin \mathcal{N}_i(t)$  keep their state variable and their companion variable constant (**RULE 3**).
- 5) Let  $t = t + 1$  and go back to step 2. ■

These interaction rules can be explained in simple words.

- *The transmitter node  $i$*  broadcasts its state value  $x_i$  to all nodes  $j \in \mathcal{N}_i$ . In doing so, it knows its out-degree and it also broadcasts the value  $z_i(t)/\delta_{out,i}(t)$  by dividing the value of the companion variable by the number of nodes that receive the information. The transmitter node  $i$  does not change its value of  $x_i(t)$  while it resets to 0 the companion variable  $z_i(t)$ .

- *The receiver nodes* update their  $x_j(t)$  variable by computing the average between their and the received state value. Furthermore they correct their update by a fraction of their companion variable  $z_j(t)$  and a fraction of the companion variable of the transmitter node  $z_i(t)$ . The receiver nodes update their companion variable by adding up several terms, designed to preserve the average of the network at each iteration while converging to the average of the initial measurements.

The sequence of nodes that perform the broadcast at the different time instants  $t \in \mathbb{N}$  defines a signal  $\mathcal{I}(t)$ . As an example if node 3 is the transmitter node at time  $t = 0$  then  $\mathcal{I}(0) = 3$ .

Assume that at time  $t$  node  $i = \mathcal{I}(t)$  performs a broadcasts, the interaction topology at time  $t$  is represented by a graph  $\mathcal{G}_i(t)$ , obtained from  $\mathcal{G}(t)$  removing all arcs whose tail is not node  $i$ .

We let  $A_i(t)$ ,  $\Delta_{in,i}(t)$  and  $\mathcal{L}_i(t)$  denote, respectively, the incidence matrix, the in-degree matrix and the Laplacian of this graph.

Let us define

$$P_i(t) = I - 0.5\mathcal{L}_i(t), \quad \hat{\Gamma}_i(t) = \frac{A_i(t)}{2\delta_{out,i}(t)} + 0.5\Delta_{in}(t),$$

$$\Gamma_i(t) = \frac{A_i(t)}{2\delta_{out,i}(t)} - 0.5\Delta_{in}(t) + (I - \mathbf{e}_i\mathbf{e}_i^T),$$

where  $I$  is the identity matrix and  $\mathbf{e}_i$  is the  $i$ -th canonical basis vector of dimension  $n$ .

We denote

$$C_i(t) = \begin{bmatrix} P_i(t) & \hat{\Gamma}_i(t) \\ I - P_i(t) & \Gamma_i(t) \end{bmatrix}. \quad (5)$$

Under the decentralized state update rule (2), (3) and (4), the system dynamics at time  $t$ , is:

$$\begin{bmatrix} \mathbf{x}(t+1) \\ \mathbf{z}(t+1) \end{bmatrix} = C_i(t) \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{z}(t) \end{bmatrix}, \quad i = \mathcal{I}(t). \quad (6)$$

**Remark 1:** In this paper it is assumed that at each instant of time each node has a strictly positive probability of broadcasting its state to the neighboring nodes. This assumption is to model the inherent asynchrony of wireless communications between sensor nodes. The results on the convergence properties of algorithms developed with this assumption hold for any deterministic scheduling of the communications between the nodes because the order in which the updates are performed is not relevant to the stability of the equilibrium point of the algorithm.

We point out that at each time instant  $t$  only the broadcaster node  $i$  has to know the number of local neighbors  $|\mathcal{N}_i|$ , which correspond to its out-degree at time  $t$ . The most trivial case in which such assumption holds is when the sensor network is not just executing a distributed average algorithm but is also providing other services that require the knowledge of the network topology. Our algorithm reduces the resources dedicated to the distributed average algorithm by not requiring acknowledgements to the transmitted information and fully exploits dense proximity networks by using broadcasts.

The main difference between the proposed algorithm and other consensus algorithms based on broadcasts is that in our approach convergence toward the exact initial average is *ensured*. In the other cases presented in the literature the consensus state is not the initial average of the measurements, but it may be any value inside the convex hull spanned by the initial conditions in the network depending on the sequence of broadcasts [17], [18].

Furthermore given that the sensor network is distributed in space, any pair of nodes sufficiently far apart can perform a broadcast while not interfering with each other. The inherent parallelism of the network is fully exploited and is expected to greatly improve the convergence time of the proposed algorithm. Nonetheless in this paper we focus our attention in studying the stability of the equilibrium point of the algorithm leaving the study of its convergence time to future work. ■

In the following, for simplicity of explanation the dependence of  $C_i$  from  $t$  will be omitted.

#### IV. ALGORITHM CONVERGENCE PROPERTIES

In this section we study the convergence properties of the algorithm. We first characterize the eigenstructure of matrices  $C_i$  and then we present a conjecture on the convergence to consensus.

**Proposition 1:**  $C_i$  is idempotent for any  $i = 1, \dots, n$ .

*Proof:* Using the general identities  $A_i^2 = A_i \Delta_{in,i} = 0$  and  $\Delta_{in,i} A_i = A_i$ , one can readily verify that for all  $i = 1, \dots, n$  it holds  $C_i^2 = C_i$ . □

Since  $C_i$  is idempotent, its eigenvalues are always either 0 or 1. Unfortunately since it is not symmetric, it represents an *oblique* projection which does not result in a contractive matrix in general.

We observe, however, that the system is conservative.

**Proposition 2:** System (6) evolves on the hyperplane

$$\mathbf{1}^T \mathbf{x}(t) + \mathbf{1}^T \mathbf{z}(t) = \mathbf{1}^T \mathbf{x}(0) + \mathbf{1}^T \mathbf{z}(0).$$

*Proof:* For all  $i = 1, \dots, n$ , the row vector  $[\mathbf{1}^T \ \mathbf{1}^T]$  is a left eigenvector for matrix  $C_i$  associated to eigenvalue 1, because it holds  $[\mathbf{1}^T \ \mathbf{1}^T] C_i = [\mathbf{1}^T (P_i + I - P_i) \ \mathbf{1}^T (\hat{\Gamma}_i + \Gamma_i)] = [\mathbf{1}^T \ \mathbf{1}^T]$ . □

Since the system is autonomous and the companion initial state can be arbitrary chosen, we select  $\mathbf{z}(0) = \mathbf{0}$ . With such assumption we obtain that the information about the average of the initial state is preserved despite communications are mono-directional and asynchronous. In particular, if there exists a time  $t$  in which  $\mathbf{z}(t) = \mathbf{0}$ , then at that time  $t$  it holds  $\mathbf{1}^T \mathbf{x}(t) = \mathbf{1}^T \mathbf{x}(0)$ .

In the following we provide an analysis of the equilibrium states of the proposed algorithm and corroborate the conjectured asymptotic stability of the equilibrium states by simulations.

Let us now consider the equilibrium points.

**Proposition 3:** The consensus state in which  $\mathbf{x}(t) = \alpha \mathbf{1}$  for some scalar  $\alpha$  and  $\mathbf{z}(t) = \mathbf{0}$  is an equilibrium state for system (6).

*Proof:* For all  $i = 1, \dots, n$ , the column vector  $[\mathbf{1}^T \ \mathbf{0}^T]^T$  is a right eigenvector for matrix  $C_i$  associated to eigenvalue 1, because it holds

$$C_i(t) \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} P_i \mathbf{1} \\ (I - P_i) \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix},$$

due to the Laplacian property  $\mathcal{L} \mathbf{1} = \mathbf{0}$ . □

We now consider the null space of the consensus matrices.

**Proposition 4:** For all  $i = 1, \dots, n$ , the kernel of  $C_i(t)$  has dimension  $\dim(Ker(C_i)) = |\mathcal{N}_i(t)| + 1$ .

*Proof:* It can be shown that the multiplicity of the null eigenvalue of  $C_i$  is  $|\mathcal{N}_i(t)| + 1$ . A set of linearly independent eigenvectors that form a basis of the null space are:

- for  $j \in \mathcal{N}_i$  a vector  $\mathbf{v}_j = [\mathbf{e}_j^T \ -\mathbf{e}_j^T]^T$ ;
- a vector  $\hat{\mathbf{v}}_i = [\hat{\mathbf{x}}_i^T \ \hat{\mathbf{z}}_i^T]^T$  with

$$\hat{\mathbf{x}}_i(j) = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise} \end{cases}$$

and

$$\hat{\mathbf{z}}_i(j) = \begin{cases} -2\delta_{out,i} & \text{if } j = i, \\ 1 & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise} \end{cases}.$$

□

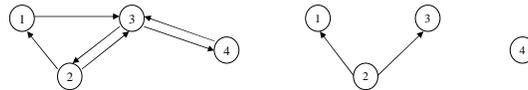


Fig. 1. Network considered in Example 1 (left). Interaction topology when node 2 performs a broadcast in Example 1 (right).

**Example 1:** Let us consider the network on the left of Fig. 1. When node 2 performs a broadcast, the interaction topology is represented by a directed graph, shown on the right of Fig. 1. The adjacency matrix for the resulting graph is  $A_2 = [0 \ 1 \ 0 \ 0; 0 \ 0 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 0 \ 0 \ 0]$ .

Following our previous definitions,  $\delta_{out,2} = 2$ , and we have:

$$\begin{aligned}
 P_2 &= [1/2 \ 1/2 \ 0 \ 0; 0 \ 1 \ 0 \ 0; 0 \ 1/2 \ 1/2 \ 0; 0 \ 0 \ 0 \ 1], \\
 \Gamma_2 &= [1/2 \ 1/4 \ 0 \ 0; 0 \ 0 \ 0 \ 0; 0 \ 1/4 \ 1/2 \ 0; 0 \ 0 \ 0 \ 1], \\
 \hat{\Gamma}_2 &= [1/2 \ 1/4 \ 0 \ 0; 0 \ 0 \ 0 \ 0; 0 \ 1/4 \ 1/2 \ 0; 0 \ 0 \ 0 \ 0];
 \end{aligned}$$

finally

$$C_2 = \left[ \begin{array}{cccc|cccc}
 1/2 & 1/2 & 0 & 0 & 1/2 & 1/4 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1/2 & 1/2 & 0 & 0 & 1/4 & 1/2 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 \hline
 1/2 & -1/2 & 0 & 0 & 1/2 & 1/4 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -1/2 & 1/2 & 0 & 0 & 1/4 & 1/2 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right].$$

By Proposition 4 the following is a basis of linearly independent eigenvectors for the null space:

$$\left[ \begin{array}{ccc}
 \mathbf{v}_1 & \mathbf{v}_3 & \hat{\mathbf{v}}_2
 \end{array} \right] = \left[ \begin{array}{ccc}
 1 & 0 & 1 \\
 0 & 0 & 0 \\
 0 & 1 & 1 \\
 0 & 0 & 0 \\
 -1 & 0 & 1 \\
 0 & 0 & -4 \\
 0 & -1 & 1 \\
 0 & 0 & 0
 \end{array} \right].$$

■

Now we consider a property that holds for strongly connected graphs.

**Proposition 5:** If

$$\hat{\mathcal{G}}[t_1, t_2] = \bigcup_{t=t_1}^{t_2} \mathcal{G}_i(t), \quad i = \mathcal{I}(t)$$

is strongly connected, then

$$\dim \left( \bigvee_{t=t_1}^{t_2} \ker(C_i(t)) \right) = 2n - 1,$$

where  $\bigvee$  denotes the linear combination of vector spaces.

*Proof:* To show this, let us take the union of all basis vectors for the null spaces of all matrices  $C_i$ , as defined in the proof of Proposition 4. Since  $\hat{\mathcal{G}}[t_1, t_2]$  is strongly connected (sufficient condition), each node is at least once a transmitter and at least once a receiver.

Thus combining all vectors we obtain the following matrix

$$V = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_n \ \hat{\mathbf{v}}_1 \ \cdots \ \hat{\mathbf{v}}_n] = \begin{bmatrix} I & A(t) \\ -I & A(t) - 2\Delta_{out}(t) \end{bmatrix}.$$

By elementary row operations we show this matrix to be equivalent to

$$\begin{bmatrix} I & A(t) \\ 0 & 2A(t) - 2\Delta_{out}(t) \end{bmatrix} = \begin{bmatrix} I & A(t) \\ 0 & -2\mathcal{L}_{out}(t) \end{bmatrix}$$

where  $\mathcal{L}_{out} = \Delta_{out} - A$  denotes the out-degree Laplacian, whose rank is  $n - 1$  if the graph is strongly connected.

Thus matrix  $V$  has rank  $2n - 1$  and this proves the result.  $\square$

Thanks to the above propositions the following important result can be proved.

**Proposition 6:** If  $\forall t > 0$  there exists  $T(t) > 0$  such that

$$\hat{\mathcal{G}}[t, t + T(t)] = \bigcup_{\tau=t}^{t+T(t)} \mathcal{G}_i(\tau), \quad i = \mathcal{I}(\tau),$$

is strongly connected, then the subspace

$$\text{span} \left( \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix} \right)$$

is the largest invariant subspace for system (6).

*Proof:* The fact that  $\text{span} \left( [\mathbf{1}^T \mathbf{0}^T]^T \right)$  is an invariant subspace was shown in Proposition 3. The fact that it is the largest invariant subspace follows from Proposition 5. In fact, the strongly connectedness of  $\hat{\mathcal{G}}[t, t + T(t)]$  implies the existence of a basis composed by vector  $[\mathbf{1}^T \mathbf{0}^T]^T$  plus a set of vectors that span  $\mathcal{K} = \bigvee_{\tau=t}^{t+T(t)} \ker(C_i(\tau))$ . Hence any vector that has initially a component along a basis vector of  $\mathcal{K}$  will have that component eventually filtered.  $\square$

We point out that the assumption of strong connectivity of the network topology *does not* require a *fixed* and constant time window  $T$  over which it is strongly connected. For Proposition 6 to hold it is sufficient that for any  $t$  there exists a  $T(t) > 0$  possibly varying but finite in which  $\hat{\mathcal{G}}[t, t + T(t)]$  is strongly connected. Note that such assumption is one of the most general assumptions that can be made regarding network connectivity because if for some time instant  $t$  there does not exist a  $T(t) > 0$  in which  $\hat{\mathcal{G}}[t, t + T(t)]$  is strongly connected, then starting from such  $t$  there exists at least a node not reachable from all the others.

Now we state the main result of this paper for which a formal proof is missing but whose relevance is shown by simulations.

**Conjecture 1:** If  $\forall t$ , there exists  $T(t) > 0$  such that

$$\hat{\mathcal{G}}[t, t + T(t)] = \bigcup_{\tau=t}^{t+T(t)} \mathcal{G}_i(\tau), \quad i = \mathcal{I}(\tau)$$

is strongly connected, if the system evolves according to the state update rule described by (6) with  $\mathbf{z}(0) = \mathbf{0}$ , then:

$$\lim_{t \rightarrow \infty} \mathbf{x}(t) = \frac{\mathbf{1}\mathbf{1}^T}{n} \mathbf{x}(0).$$

■

The above conjecture is validated by several numerical experiments, some of which are reported in the following simulations section, and by the following theoretical observations.

First, by Proposition 3, it holds that  $[\alpha \mathbf{1}^T \mathbf{0}^T]^T$  is an equilibrium state for system (6) for some scalar  $\alpha$ . Secondly, by Proposition 2, being by assumption  $\mathbf{z}(0) = \mathbf{0}$ , it holds that  $\mathbf{1}^T \mathbf{x}(t) + \mathbf{1}^T \mathbf{z}(t) = \mathbf{1}^T \mathbf{x}(0)$  for any  $t$ . Finally, by Proposition 6,  $\text{span} \left( [\mathbf{1}^T \mathbf{0}^T]^T \right)$  is the largest invariant subspace for system (6).

Unfortunately, the problem of deciding whether the random product of a finite set of matrices converges is still an open problem in matrix theory and all the results are either not applicable or relate to classes of matrices not suitable for our purposes.

The proposed conjecture, while intuitive and validated by simulation results poses great difficulties in its proof. First, the description of the stability of a switching linear system of the type of system (6) has been treated only for simple cases in which at each instant of time  $t$  the system matrix is at least para-contractive. Others have used Markov chain theory and applied

it to the study of consensus problems. Some other results use the Common Lyapunov function approach to study the convergence properties of such systems. In our case the system matrix  $C$  is not symmetric, is not contractive, nor is it with non-negative elements thus invalidating almost all known general results for stability analysis of such system.

## V. SIMULATIONS

In this section we provide simulations in order to corroborate the algorithm analysis.

Each node is initialized with values between 0 and 1 chosen at random from a uniform distribution. The simulations are scaled respect to the number of nodes in the network for topologies whose features do not change as the number of nodes is increased.

We now define an error performance index:

$$V(t) = \|\mathbf{x}(t) - \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{x}(0)\|_2 + \|\mathbf{z}(t)\|_2.$$

Such index is a measure of how far the state of the network is from its equilibrium state, namely

$$\mathbf{x}(\infty) = \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{x}(0), \quad \mathbf{z}(\infty) = 0.$$

The convergence time is function of the network topology and the number of nodes. Let us define the convergence time proposed in the simulations as

$$T_{con} = \inf\{\tau > 0 : \frac{V(\tau)}{V(0)} < 0.05\}, \quad (7)$$

i.e. the number of broadcasts needed such that the error modeled by the performance index  $V(t)$  becomes less than 5% of its value at initialization time. Note that, since  $V(t)$  is not a non-increasing function of  $t$ , the fact that  $V(t)$  is less than 5% of its initial value at a given time  $t$ , does not imply that it remains less than this value for all  $\tau > t$ .

The following simulations show the average value of  $T_{con}$  over 100 realizations of the algorithm. We consider three different topologies:

- *Fully connected topology*: each broadcast is received by every node in the network.
- *Line topology*: each broadcast is received only by the 2 adjacent nodes of the broadcaster except for the one of the end nodes.
- *Square grid topology*: each broadcast is received only by those nodes in proximity of the broadcaster in a square grid.

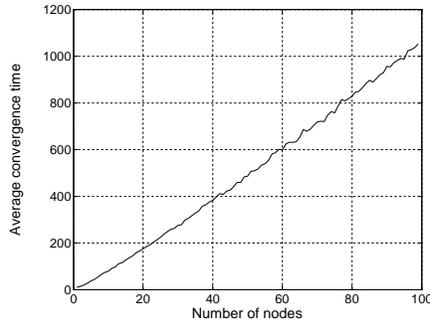


Fig. 2. Average value of  $T_{con}$  for a fully connected network with respect to the number of nodes.

The selected topologies specify the edges that *may* be selected by the gossip algorithm at any given time instant. We simulated the algorithm on regular and well defined graphs topologies so that the presented simulations results are easily reproducible. Furthermore by selecting well defined topologies we can easily compute average convergence times when transmitter nodes are selected at random. On the contrary, in the case of random topologies the convergence time not only would vary depending on the edge selection process but also due to the given realizations of the random topologies, thus greatly increasing the variance of the convergence time and impairing the clarity of the results.

Fig. 2 is relative to simulations performed on fully connected networks with an increasing number of nodes. Simulations show that the average convergence time scales linearly with the number of nodes as  $T_{con} \approx 10n$ .

In Fig. 3 the simulations are performed for line networks of increasing number of nodes. Simulations show that the convergence time scales polynomially respect to the diameter of the graph which is equal to  $n - 1$  for line graphs.

In Fig. 4 the simulations are performed for square grid networks of increasing number of nodes, such that the total number of nodes is perfect square. Simulations show that the convergence time scales polynomially respect to the diameter of the graph which is equal to  $\sqrt{n} - 1$  for grid graphs.

The improvement of the proposed algorithm respect to other gossip algorithms is that by using broadcasts the inherent parallelism in a distributed network is fully exploited between all the nodes and not only between nodes not directly connected. This feature is especially relevant in networks with small diameter where few nodes have a very high out-degree such as small world

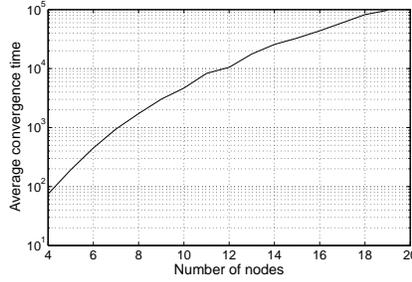


Fig. 3. Average value of  $T_{con}$  for a line network with respect to the number of nodes.

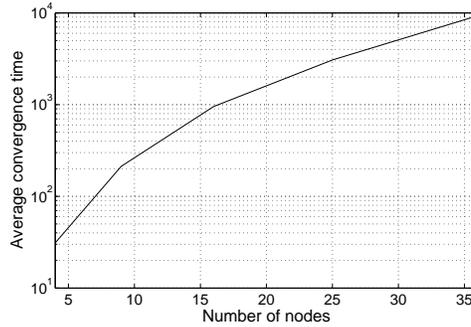


Fig. 4. Average value of  $T_{con}$  for a square grid network with respect to the number of nodes.

networks.

### *A comparison with the Standard Gossip based on Broadcast*

We now compare the proposed algorithm with a simple gossip algorithm illustrated in [17], [18] without our average preserving properties. In these works a gossip algorithm based on broadcasts is studied which can be summarized as follows.

**Algorithm 2** (*Standard Gossip based on Broadcasts (SGB)*):

- At each instant of time a node broadcasts its value to its neighbors.
- If at any time a node listens to a broadcast, it computes the average between its state and the broadcaster state. It then takes this new value as its state.
- Repeat until all the nodes have the same value. ■

**Remark 2:** The following numerical simulations compare our algorithm with the SGB algorithm [17], [18] in terms of convergence rate. However, the main difference among the two

algorithms is *qualitative*: the proposed algorithm converges *exactly* to the average of the initial states while the SGB algorithm may converge to any point inside the convex hull spanned by the initial network conditions depending on the sequence of broadcasts. It has been shown that, if the number of nodes is sufficiently high and broadcasts occur at random, then the SGB converges *statistically* sufficiently close to the initial average. Indeed, the main point of adding more sensors to a network to measure the same scalar quantity is to get out of cheap sensors a measurement whose precision is far greater than the precision of the single units but this can be achieved through averaging only if the network does actually converge to the initial average. ■

Now, since the SGB algorithm does not converge to the average, we define a different performance index which becomes zero only when each node has the same value:

$$\hat{V}(t) = \|\mathbf{x}(t) - \frac{\mathbf{1}\mathbf{1}}{n}\mathbf{x}(t)\|.$$

In the simulations of Algorithm 2 we adopt the following definition of convergence time:

$$\hat{T}_{con} = \inf\{\tau > 0 : \frac{\hat{V}(\tau)}{\hat{V}(0)} < 0.05\}. \quad (8)$$

To corroborate the simulation results, we show the average error over the 100 realizations respect to the initial average of the network at  $t = \hat{T}_{con}$ , namely

$$Err(\hat{T}_{con}) = \frac{\|\mathbf{x}(\hat{T}_{con}) - \frac{\mathbf{1}\mathbf{1}}{n}\mathbf{x}(0)\|}{n}.$$

The simulations are again performed for the three topologies taken into consideration:

- Simulations on a *Fully connected topology* are shown in Fig. 5. The convergence time results to be constant when increasing the number of nodes. This can be explained by the fact that being the network fully connected, at each broadcast all the nodes receive information disregarding the size of the network. In Fig. 6 the average error at convergence time is shown for the same simulations: the error decreases as more nodes are added.

- Simulations on a *Line topology* are shown in Fig. 7. The convergence time appears to be polynomial in the number of nodes and an order of magnitude lower than our proposed algorithm. In Fig. 8 is shown the average error at convergence time for the same simulations: also in this case adding more nodes reduces the final average error.

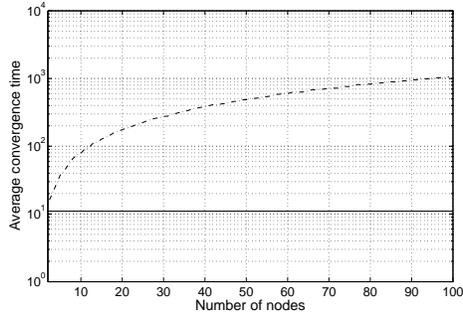


Fig. 5. Average value of  $T_{con}$  (dashed line) and  $\hat{T}_{con}$  (continuous line) for a fully connected network with respect to the number of nodes.

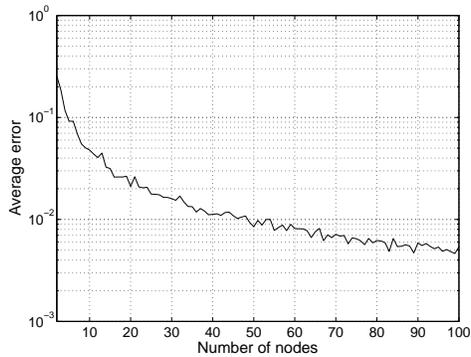


Fig. 6. Average value of  $Err(\hat{T}_{con})$  for a fully connected topology using the SGB algorithm.

- Simulations on a *Grid topology* are shown in Fig. 9. The convergence time appears to be polynomial and faster than our proposed algorithm. In Fig. 10 is shown the average error at convergence time for the same simulations: again it decreases when increasing the number of nodes.

Summarizing, simulations show that the SGB algorithm achieves better convergence times for growing network size trading off a finite error in the final state that decreases with network size. Such trade-off is the highest when the number of nodes is small.

#### *A comparison with standard Gossip based on pairwise averaging*

We now compare our algorithm with the standard gossip algorithm based on random pairwise averaging (SGP) [8], [9], [10], [11]. We believe that comparing the performances of these two

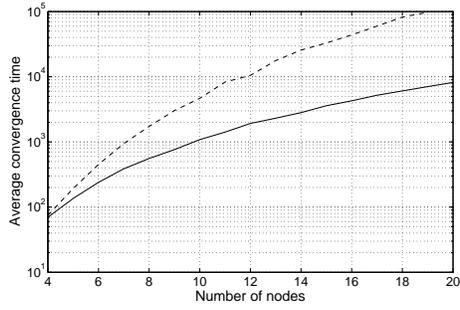


Fig. 7. Average value of  $T_{con}$  (dashed line) and  $\hat{T}_{con}$  (continuous line) for a line topology with respect to the number of nodes.

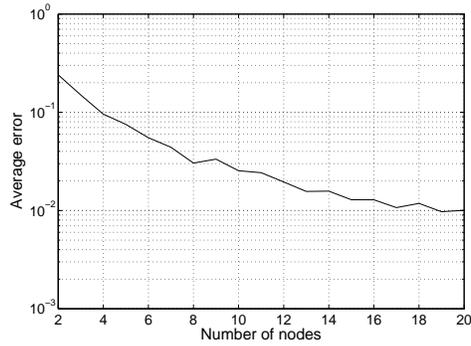


Fig. 8. Average value of  $Err(\hat{T}_{con})$  for a line topology using the SGB algorithm.

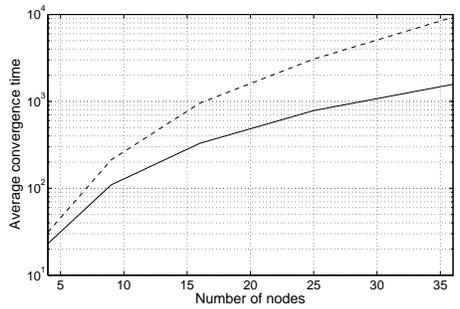


Fig. 9. Average value of  $T_{con}$  (dashed line) and  $\hat{T}_{con}$  (continuous line) for a square grid topology with respect to the number of nodes.

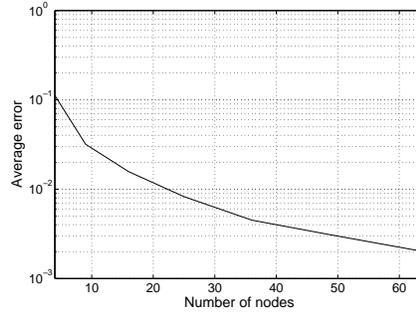


Fig. 10. Average value of  $Err(\hat{T}_{con})$  for a square grid topology using the SGB algorithm.

algorithms is significant because they both converge exactly to the average of the initial state of the network.

A gossip algorithm based on pairwise communications can be summarized as follows.

Let  $\mathcal{G} = \{V, E\}$  represent the network topology at time  $t$ . Let  $x(t)$  be an  $n$ -element vector representing the state of the network where the generic element  $x_i(t)$  is the state of node  $i$ . Let  $\mathcal{E}(t) : \mathbb{R}^+ \rightarrow E$  be the edge selection process that at any given time instant outputs an edge  $e_{ij} \in E$ .

**Algorithm 3** (Standard Gossip based on pairwise averaging):

- Let  $t = 0$  and  $x(t) = x_0$  be the initial state of the network.
- Select the edge  $e_{ij}$  given by the edge selection process  $\mathcal{E}(t)$ .
- Let

$$\begin{cases} x_i(t+1) = \frac{x_i(t) + x_j(t)}{2}, \\ x_j(t+1) = \frac{x_i(t) + x_j(t)}{2}. \end{cases}$$

- Let  $t = t + 1$  and go back to step 2. ■

**Remark 3:** Before presenting the results of numerical simulations, let us discuss some qualitative difference between the two algorithms: the proposed algorithm works on arbitrary strongly connected directed graphs while the SGP algorithm requires bidirectional information exchange at each time step and thus can only be applied to connected undirected graphs. For this reason we can compare the performance of the two algorithms only for those restricted network topologies that satisfy the connectivity requirements of the SGP algorithm.

Moreover, several gossip algorithms based on pairwise averaging differ in the edge selection

Broadcast radius	Transmission power	Convergence time		Total Energy	
		SGP Algorithm	EGB Algorithm	SGP Algorithm	EGB Algorithm
1.10	1.23	1406	9405	1406	9406
1.57	2.43	931	3948	2095	8883
2.10	4.41	680	1271	2723	5086
2.97	8.82	528	421	4445	3544
3.10	9.61	494	341	4446	3077
4.10	16.81	445	253	7133	4059
4.39	19.27	451	241	8352	4455
5.10	26.01	426	227	10659	5684
5.80	33.64	434	236	14100	7678
6.10	37.21	433	233	15592	8391
7.22	52.13	434	228	21894	11511
8.63	74.48	426	232	30809	16810

TABLE I  
SIMULATION ON A 49 NODE GRID PROXIMITY NETWORK.

process, for instance in [8] the nodes are selected according to a Poisson process. In general, the edge selection process affects the convergence time in absolute terms, while the sequence in which edges are chosen affects the number of iterations required, which indirectly affects the convergence time. To perform a single iteration of the GSP algorithm requires two transmissions, node  $i$  needs to send its state value to node  $j$  and node  $j$  has to reply to node  $i$  with its own state. In this paper we neglect the complexity of data link and physical layers because, even with this crude simplification of the bidirectional communication process, we can show the advantage of our algorithm respect to the SGP algorithm in terms of the total energy required to achieve a given performance. ■

In the following simulations a set of 49 wireless sensors has been placed in a  $7 \times 7$  grid spaced 1 unit of length  $[m]$  between each other. The network has been simulated by varying the total transmitted power of each node to compute how the energy consumed by the algorithms scale with the transmitted power on proximity graphs. Note that we take into consideration a grid network and not the more general set of random geometric graphs, to eliminate the contribution

of the randomness of the topology to the energy spent during the algorithm execution, which is already random due to the particular edge selection process.

In these simulations we take inspiration from [11] that as in our case simulates changes in network topologies due to increases in transmitted power. Such changes of the communication range are proportional to the square root of the transmitted power taking into account total communication costs in the performance index.

To model the proximity range as function of the transmitted power we consider the standard equation for radio frequency communications:

$$P_r = \frac{P_t g_t g_r}{4\pi r^2}$$

where  $P_t$  and  $P_r$  are, respectively, the transmitted and received power in watt [W],  $g_t$  and  $g_r$  are the antenna gains of the receiver and the transmitter and  $r$  is the line of sight distance between transmitter and receiver. The receiver, given the technological constraints due to the electronics and noise, needs to receive at least a given power  $P_r$  to be able to decode the message sent.

In our simulations we simplify this model by allowing the effective communication radius  $r$  to scale as  $r \propto \sqrt{P_t}$ . The topology of the grid network then requires at least  $P_t = 1$  to be connected, i.e. to allow each node to communicate bidirectionally with at least one neighbor. Increasing the transmitted power the number of neighbors that each node can communicate with increases until the transmitted power is enough to reach any node in the network, roughly for  $r = 6\sqrt{2}$  and  $P_t = r^2 = 72$ .

Each transmission is assumed to last  $\tau$  seconds and so consume  $E = P_t \tau$  [J] units of energy. Again we simplify the simulation assuming  $\tau = 1$  since the improvement of the proposed algorithm respect to the SGP algorithm is the reduction of total number of transmissions and does not depend on  $\tau$ .

The execution of the algorithms is stopped as soon as the performance index

$$\tilde{V}(t) = \|\mathbf{x}(t) - \frac{\mathbf{1}\mathbf{1}^T}{n}\mathbf{x}(0)\|_2$$

reaches 10% of its initial value, i.e., we define the average convergence time as

$$\tilde{T}_{con} = \inf\{\tau > 0 : \frac{\tilde{V}(\tau)}{\tilde{V}(0)} < 0.1\}, \quad (9)$$

In Fig. 11 and 12 a comparison between the proposed algorithm (continuous line with square

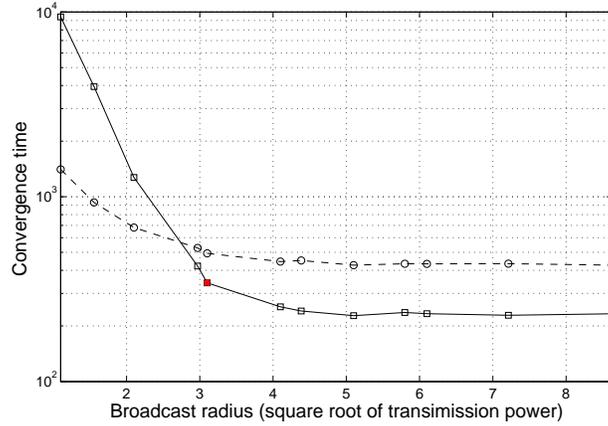


Fig. 11. Total number of transmissions executed by the SGP algorithm (dashed line) and the proposed algorithm (continuous line) respect to the broadcast radius for a grid of sensors.

markers) and the SGP algorithm (dashed line with round markers) is shown. In particular, in Fig. 11 the total number of transmissions is plotted against the broadcast radius which affects the topology by increasing the number of neighbors of each agent in the grid. In Fig. 12 the total energy consumption is shown respect to the transmitted power at the nodes.

It can be seen that, when the number of neighbors of each node is small due to little transmitted power, the SGP algorithm performs better both in number of transmissions and energy saving. This is reasonable because there is no gain in performing broadcasts if the number of nodes that can listen to it is very small. As soon as the out-degree of each node increases for increasing values of the broadcast radius, the performance of the proposed algorithm based on broadcasts becomes significantly better of almost one order of magnitude in number of transmissions and energy saving. In particular, it is clear that the proposed algorithm achieves a better performance, in the case of the grid topology, achieving a minimum-energy consumption when the broadcast radius equals 3.1, corresponding to the case in which each node can reach other nodes distant at most 3 rows or columns in the grid network (sample marked in red). The results of the above numerical simulations are also summarized in Table I.

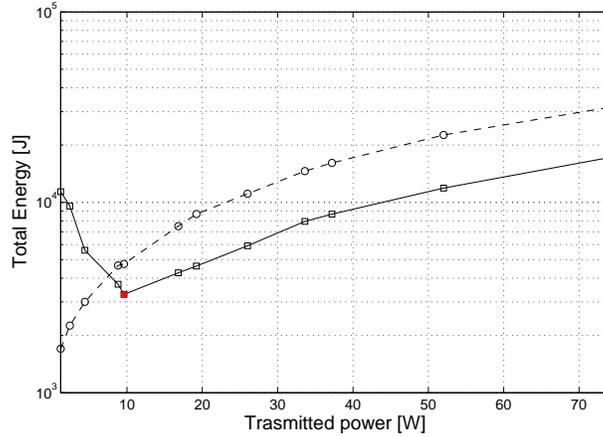


Fig. 12. Total energy in [J] consumed by the execution of the SGP algorithm (dashed line) compared with the proposed algorithm based on broadcasts respect to normalized transmitted power [W] for a grid of sensors.

## VI. CONCLUSIONS

In this paper we have proposed a novel gossip algorithm based on broadcasts that achieves consensus on the average on arbitrary strongly connected digraphs. The study of the convergence properties is preliminary and convergence is shown by simulations. The proposed algorithm is based on gossip and preserves the information about the average of the initial state during its execution. The main feature of our algorithm is that it converges *exactly* to the average of the initial state. Moreover, it works on arbitrary strongly connected digraphs.

A comparison with the standard gossip algorithm based on broadcast and with the standard gossip based on pairwise averaging has also been made. Simulations show that the proposed algorithm achieves better convergence rates and energy saving than the standard gossip based on pairwise averaging if the number of neighbors of each node is sufficiently high.

## REFERENCES

- [1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.
- [2] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [3] R. Olfati-Saber, "Flocking for multi-agent dynamic system: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

- [4] X. Lin and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [5] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *Information Theory and Applications Workshop*, pp. 150–159, 2007.
- [6] M. Porfiri and D. Stilwell, "Consensus seeking over random weighted directed graphs," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1767–1773, 2007.
- [7] G. Picci and T. Taylor, "Almost sure convergence of random gossip algorithms," in *46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007.
- [8] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [9] M. Rabbat, "On spatial gossip algorithms for average consensus," in *IEEE/SP 14th Workshop on Statistical Signal Processing, Madison, Wisconsin, US*, 2007, pp. 705–709.
- [10] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, 2009.
- [11] —, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7 Part 2, pp. 3315–3326, 2008.
- [12] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, 2007.
- [13] U. A. Khan and J. M. F. Moura, "Distributed kalman filters in sensor networks: Bipartite fusion graphs," in *IEEE/SP 14th Workshop on Statistical Signal Processing, Madison, Wisconsin, USA*, 2007.
- [14] M. Alighanbari and J. How, "An unbiased kalman consensus algorithm," in *American Control Conference, Minneapolis, Minnesota, USA*, 2006.
- [15] M. Franceschelli, A. Giua, and C. Seatzu, "Consensus on the average on arbitrary strongly connected digraphs based on broadcast gossip algorithms," in *1st IFAC Workshop on Estimation and Control of Networked Systems*, Venice, Italy, Sep. 2009.
- [16] I. Daubechies and J. Lagarias, "Sets of matrices all infinite products of which converge," *Linear algebra and its applications*, vol. 161, pp. 227–263, 1992.
- [17] T. Aysal, M. Yildiz, A. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," in *IEEE Transactions on Signal processing*, vol. 57, no. 7, 2009, pp. 2748–2761.
- [18] —, "Broadcast gossip algorithms: Design and analysis for consensus," in *47th IEEE Conference on Decision and Control, Cancun, Mexico*, 2008, pp. 4843–4848.