An Approach for Weighted Average Consensus in Event Detection*

Cristian Dragana[1], Grigore Stamatescu[1], Viorel Mihai[1], Dan Popescu[1]

Abstract— **Large scale monitoring systems require reliable and efficient in-network information extraction mechanisms able to effectively track events at the field level. The study of consensus algorithms for distributed data processing has gained a lot of interest in the last decade. Average consensus algorithms used for decentralized sensor fusion in wireless sensor networks, iteratively compute the global average value, in a completely distributed manner through local information exchange among neighbors. In the first instance, it is mandatory to pursue the reduction of convergence time, for energetic reasons, but it is also essential to lead the convergence to a reliable value. In this paper we propose a new weighted average consensus algorithm, tailored for event detection where each sensor selects its own weights on the basis of some local information regarding number of direct neighbouring nodes and estimated distances to each neighbour. Various simulations have been implemented and analysed from a comparative standpoint.**

I. INTRODUCTION

In the last decade, in-network data processing has gain a lot of interest, mainly due to the rapid development of embedded devices. The increased adoption of WSNs (Wireless Sensor Networks) across numerous applications with high prevalence in agriculture, critical infrastructure, industries, mainly in oil and gas and waste water and even military surveillance missions, is due to some already well known reasons [1][2]. Among these strong reasons, main benefits rely to ease of implementation (no cable run), able to operate in different environments, scalability and robust technology with surprising computation capabilities. Moreover, under the rapid ongoing proliferation of IoT (Internet of Things), WSNs has become essential elements for making the most of this [3]. We build upon previous work in [4] where we proposed a novel approach for smart data collecting in large areas based on WSN and Unmanned Aerial Vehicle (UAV) collaboration.

Large monitoring systems based on WSN systems require efficient decentralized sensor fusion algorithms able to perform consensus decision making in a way that leverages the on-board computing resources of each sensor node and reduces the burden on the communication channel. Sensor fusion schemes using in-network data processing, based on consensus algorithm, provide fault tolerant solutions for event detection applications. In a consensus algorithm, each node estimates the global average in a linear iterative way through local information exchange, governed by the interconnection topology. Each node updates its own estimation based on local knowledge and an updating rule. One of the main concerns regarding WSN nodes refer to energy consumption, specific to battery operated embedded devices. Thus, it is mandatory to pursue the reduction of convergence time, by reducing the number of consensus steps. Convergence speed of average consensus mechanism can be improved using a weighted updating rule. In theory, weights should be selected stemming from the network topology and some basic statistics, related to graph theory, such as connectivity or maximum degree. Due to the restriction regarding data transfer among sensor nodes, for energy saving, in practice, well known average consensus algorithms use only local neighbor information (regarding the topology) for the weights selection.

In certain scenarios, estimating the global average value is not sufficient to detect the occurrence of an event. The emergence of isolated deviations can be iteratively suppressed by overall average related to the value with the highest incidence. Such behavior should not be tolerated in certain event detection application where the notification of initial deviations in an early stage can bring important savings or benefits.

In this paper we propose a new weighted average consensus algorithm, tailored for event detection where each sensor selects its own weights on the basis of some local information regarding number of direct neighbouring nodes and estimated distances to each neighbour. Distances are estimated by means of signal quality evaluation stemming from indicators such as RSSI (Received Signal Strength Indicator) or LQI (Link Quality Indication).

The outline of the paper is as follows. Section II provides a brief overview of average consensus algorithms and further explores some well known methods for weights selections. Section III presents our approach for weighted average consensus, tailored for event detection systems. The algorithm is suitable for static clustered WSNs with synchronous local information exchange. In Section IV, simulation results are presented with focus both on convergence time and consensus accuracy. Last section summarizes the paper and discusses future work.

II. AVERAGE CONSENSUS ALGORITHMS

In a consensus method, multiple autonomous agents seek to reach a collective goal under the influence of the information flow in the network.

WSNs are usually modeled as graphs with directed or undirected edges, corresponding to the allowed information flow between network nodes. In literature there are several studies concerning the consensus paradigm in multi-agent

*This work has been been funded by the University “Politehnica” of Bucharest, through the ”Excellence Research Grants” Program, UPB-GEX. Identifier: UPB-EXCELENTA-2016 ”New Dynamic Consensus Algorithms for Building Thermal Energy Management (CONSENSYS), Contract number 20/26.09.2016.

[1]Department of Automatic Control and Industrial Informatics, University ”Politehnica” of Bucharest, 060042 Bucharest, Romania

system related to algebraic graph theory. A good reference is [5].

In order to describe consensus methods, we assume an undirected graph representation of the information flow among the network nodes, formal denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the vertices $\mathcal{V} = 1, 2, \ldots, N$ correspond to the sensor nodes and the set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent the communication links between two autonomous nodes. Further we denote the elements of $\mathcal{E}$ with (i, j) where nodes i and j are able to exchange information. Since we refer to undirected graphs, the converse $(j, i) \in \mathcal{E}$. Let $A = [a_{ij}]$ be the adjacency matrix of graph $\mathcal{G}$. We denote the set of neighbours of node i with N_i defined as [6]:

$$N_i = \{j \in \mathcal{V}: a_{ij} \neq 0\}. \tag{1}$$

A distributed consensus algorithm is described in [7] by the following conventional linear system in a continuous time model:

$$\dot{x}_i(t) = \sum_{j \in N_i} a_{ij} \left(x_j(t) - x_i(t)\right) \tag{2}$$

where $x_i \in \mathbb{R}$ denote the estimated value of node i. Consensus is reached if $x_i = x_j$ for all nodes i, j where $i \neq j$.

In a discrete-time representation, the consensus algorithm is expressed as:

$$x_i[k+1] = \sum_{j \in N_i} a_{ij} \left(x_j(k) - x_i(k)\right). \tag{3}$$

Graph Laplacians are specific graph theory matrices that play an important role in analyzing the convergence of consensus algorithms and their performances. Thus, the system model (3) is expressed in the following compact representation:

$$\dot{x}_i = -Lx, \tag{4}$$

where L denotes the graph Laplacian of $\mathcal{G}$. The graph Laplacian is defined as the subtraction:

$$L = D - A, \tag{5}$$

where the diagonal matrix D is the degree matrix with $d_i = \sum_{i \neq j} a_{ij}$ diagonal elements.

It has been recognized that the properties of Laplacian matrix provide useful instruments in analysis of convergence of consensus algorithms and guiding towards a desired outcome. For undirected graphs, L is a symmetric matrix with real eigenvalues, in an ascending order as:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2\Delta, \tag{6}$$

where Δ is the maximum degree of the graph. The second smallest eigenvalue of L, denoted λ_2 and known as algebraic connectivity of the graph is used as a measure of convergence performance [8]. Several works discuss the problem of distributed estimation of the eigenvalues of Laplacian matrix [8]-[11].

In this paper, we are particularly interested in weighted average consensus methods.

In case of weighted average consensus, the conventional expression of consensus algorithm (2) becomes:

$$\omega_i \dot{x}_i(t) = \sum_{j \in N_i} a_{ij} \left(x_j(t) - x_i(t)\right), \tag{7}$$

where ω_i belongs to the desired weighting vector $\omega = (\omega_1, \omega_2, \cdots, \omega_n)$. In a compact representation, the updating rule becomes:

$$W\dot{x}_i = -Lx, \tag{8}$$

with W the weight matrix with the same pattern as the adjacency matrix.

Although the strong background stemming from algebraic graph theory provides support for the synthesis of advanced optimization procedures, some constraints specific to WSNs causes it to become unfeasible. For large sensor networks, whole topology discovery requires advanced computation complexity and involves enormous waste of energy. Thus, it seems more promising to use local average consensus methods, i.e. solution that charges each sensor to select its own weights on the basis of some local information.

An update law for node n_i based on local weighted consensus in a discrete-time representation [12] is defined as:

$$x_i(k+1) = \omega_{ii} x_i(k) + \sum_{j \in N_i} \omega_{ij} x_j(k), \tag{9}$$

where ω_{ij} is the weight elected by node i for the value of node j, and ω_{ii} denotes the weight applied to its own estimated value.

In matrix form this is expressed as:

$$x(k+1) = Wx(k), \tag{10}$$

where W is a matrix containing ω_{ij} weights elements and $x(k)$ is the state vector at convergence step k.

Some of the specific methods, which are often considered a strong basis for comparison purpose, are further described in this section.

The Metropolis weight algorithm is a method that computes the weights stemming from the degree of neighbours. This method is defined as:

$$\omega_{ij} = \begin{cases} \frac{1}{max\{d_i, d_j\}} & if\ (i,j) \in \mathcal{E}, i \neq j \\ 1 - \sum_{j \in N_i} \omega_{ij} & i = j \\ 0 & if\ (i,j) \notin \mathcal{E}, i \neq j \end{cases}, \tag{11}$$

where d_i and d_j are the degrees of nodes i, respectively j.

A particular form of this method is the Metropolis-Hasting weighting algorithm, which defines a set of weights for immediate neighbours $j \in N_i$ as following:

$$\omega_{ij} = \frac{1}{1 + \frac{1}{max\{d_i, d_j\}}}\ if\ (i,j) \in \mathcal{E}, i \neq j \tag{12}$$

The Maximum Degree weighting method relies on global knowledge, which makes it unfeasible for large monitoring systems.

It is supposed that all the nodes know the maximum node degree in the network and select the same weight value as:

$$\omega_{ij} = \begin{cases} \frac{1}{d_{max}} \; if \; (i,j) & \in \mathcal{E}, i \neq j \\ 1 - \sum_{j \in N_i} \omega_{ij} & i = j \\ 0 \quad if \; (i,j) & \notin \mathcal{E}, i \neq j \end{cases} \tag{13}$$

In [13], [14] it has been shown that optimal constant weight values are achieved by Best constant method, which is defined as:

$$\omega_{ij} = \begin{cases} \frac{2}{\lambda_1(L)+\lambda_{n-1}(L)} & if \; (i,j) \in \mathcal{E}, i \neq j \\ 1 - \sum_{j \in N_i} \omega_{ij} & i = j \\ 0 & if \; (i,j) \notin \mathcal{E}, i \neq j \end{cases}, \tag{14}$$

where $\lambda_i(L)$ is the i^{th} eigenvalue of the graph Laplacian matrix.

In [15] the authors propose a new local average consensus, called Neighborhood algorithm. This method is tailored for networks described by clustered structures and pursues to find the most critical links between sensor clusters to assign them the highest weights. This solution outperforms conventional algorithms in terms of convergence speed. A novel corrective consensus algorithm was introduced in [16] through the addition of a measure of amount of change, estimated between convergence steps. The authors define two types of iterations: standard and corrective. The corrective iteration is performed to adjust de state variables and auxiliary variables. It was shown that through this method, consensus value converges almost to the average, notwithstanding asymmetric link losses. In [17] the consensus problem of multi-agent systems with distance-dependent communication networks is discussed. It is assumed that the communication weight between two nodes is a non-increasing function of their distance.

III. Proposed Algorithm

In this paper we are mainly focused on a novel approach for weighted average consensus, tailored for event detection. The proposed solution aims at correcting the convergence value or rather guide it to an expected value. In some specific scenarios, certain highly localized events can easily pass undetected. To avoid this issue, a mechanism for detecting deviations in relation to local neighborhood average is required.

The proposed approach is suitable for static topologies, with synchronous in-network data processing. This assumes that each and every sensor node, comprising the WSN, has to be synchronized with a global clock. Thus, all the sensing nodes assist each convergence iteration. Although our work refers only to static topologies, the proposed solution may be extended for the study of dynamic WSN topologies.

Stemming from the strong basis of algebraic graph theory, the proposed solution is build on a weighted average consensus algorithm base, using a specific mechanism for weights selection. Each sensor node computes its weights, using an estimation of the distance to each neighbor and additional local information regarding the average value around its neighborhood. This mechanism ensures high priority for the sensor nodes in the proximity of the isolated deviations.

In order to estimate the distance to a neighbor, the node evaluates the quality of the received messages through some particular metrics such as the Received Signal Strength Indicator (RSSI) value (in dBm) of the unicast messages from the neighbours. Based on this estimation, the algorithm ensures the updating rule computes the current convergence value keeping a high priority for the closest neighbours. This provides asymptotic consensus convergence, and follows the average value of the neighbourhood. Weights applied to each neighbour are computed as the ratio between the closest estimated neighbour and the estimated distance as follows:

$$\omega_{ij} = \begin{cases} \frac{d_{min}}{d_{ij}} & if \; (i,j) \in \mathcal{E}, i \neq j \\ 0 & if \; (i,j) \notin \mathcal{E}, i \neq j \end{cases}, \tag{15}$$

where d_{ij} denotes the estimated distance between node i and its neighbor j and d_{min} is the distance to the closest neighbour. It can be observed that nodes outside the neighbourhood are not considered.

Using the selected weights, the algorithm performs a weighted average of neighbours values defined as:

$$N_i mean(k+1) = \frac{\sum_{j \in N_i} \omega_{ij} x_j(k)}{dim(N_i)}, \tag{16}$$

where $dim(N_i)$ is the number of neighbors.

To give high priority to sensor nodes with deviated measured values, we added additional weights for the updating rule. For the last step of the convergence iteration, the algorithm computes the following update:

$$x_i(k+1) = \frac{\phi_{ij}(k+1) \cdot x_i(k) + \varphi \cdot N_i mean(k+1)}{\phi_{ij} + \varphi}, \tag{17}$$

where the weight φ represents a predefined corrective factor in strong correlation with the dynamics of the convergence and the weight ϕ_{ij} defined as:

$$\phi_{ij} = \left| |x_i(k) - N_i mean(k+1)| - \varphi \right|. \tag{18}$$

As one can see, the weight applied to the state value is computed for each step of the average consensus and ensures a higher priority for the deviated measured values which relies to isolated emergence of events.

We further analyze the performance of the proposed algorithm from a comparative standpoint, both from different simulations and real implementation scenario.

IV. Simulation and Results

To analyze the performance of this approach, we first performed a number of comparative simulations in order to achieve the best results, stemming from the selection of the corrective factor through a heuristic. Due to the early-stage of this research topic, for improved event detection mechanism, we further describe preliminary results for some specific situations.

Note that for all the simulations, including the real implementation of the weighted average consensus algorithm, we used the same configuration. It is a static topology described by an undirected graph, comprising ten sensor nodes, in a random deployment, with a clustered structure and predefined measured values.

A comparative view of the corrective factor influence on the convergence dynamics is illustrated in Figure 1. Two deviated values (over *500* on the scale) were added in order to simulate an isolated event. As one can see the initial values update in an iterative way and slowly converge to the average value. It can be seen the strong correlation of the corrective factor and the convergence dynamics. An increasing of this weight will speed up the convergence, but the final average will be less influenced by the deviated values. This should be carefully weighted in order to obtain satisfactory results.

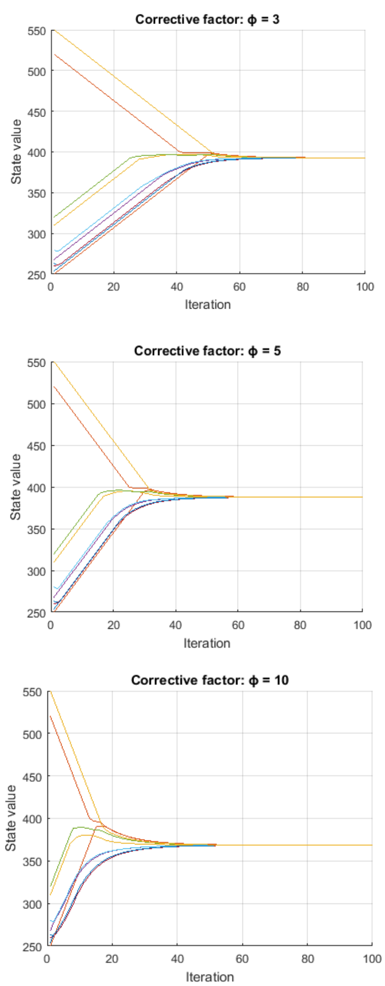


Fig. 1: Running consensus on a 10-node random topology, with undirected links, using different corrective factor values (3, 5 and 10 respectively)

In order to validate this solution for improved event detection, we performed a comparative evaluation. We took advantage of the new available virtual tools for WSN prototyping. This section showcases some of our experimental results drawn from real test-beds simulations. Our experiments are based on TelosB/Tmote Sky platforms compatible with Contiki OS. Contiki is a state-of-the-art, open source operating system for WSN, Internet of Things (IoT) and other network embedded devices.

Ten sensor nodes were deployed in a random topology, as we mentioned before. Among these, eight were initialized with values uniformly distributed in the range 250:320. The other two remaining received deviated values, as in case of the emergence of an isolated event, with values ranging 520:550. Fig. 2 illustrates the topology of the simulated WSN, together with the initial values associated to each sensing node.

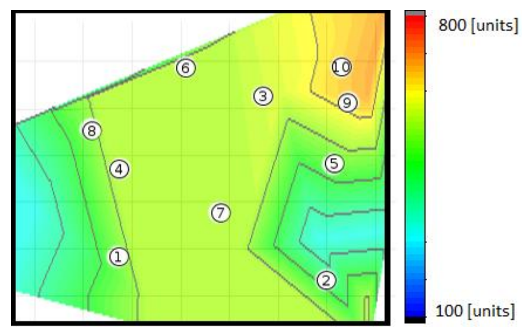


Fig. 2: WSN deployment and initial values (Heat-Map)

As one can see, the nodes were randomly deployed and the initial values were generated for a specific scenario of an isolated event occurrence. This is captured in the “Heat-map” representation, located in the right corner of the representation (sensor nodes 10 and 9).

The convergence mechanism is based on RIME communication massages exchanged between neighbour nodes in a random manner, using unicast messages. As we mentioned in the previous section, our approach is based on synchronous computation. In the first instance, each and every sensor node performs a neighbourhood recognition mechanism, based on sending/receiving broadcast messages in a randomly way. Thus, each sensor node defines a list of neighbours.

The distance estimation is based on a standard receiving quality signal indicator – RSSI. This is provided by the operating system for each message received.

The convergence algorithm is described by the following steps:

Algorithm: Weighted Average Consensus

1. If a random timer (with a predefined base time) expires, node i sends a Unicast message to a random neighbour comprising its own current value.
2. If a unicast is received, the current value of the sender is updated in the neighbour list, together with the RSSI

value and the sender is marked as received.

3. After receiving all the values from its neighbours, node i assigns to each link $(i,j), j \in N_i$ the corresponding weight related to the max(RSSI).
4. The weighted average of the neighbour values is computed according to relation (16) described in section III.
5. The weight ϕ_{ij} is computed based on relation (18).
6. Compute the updating rule described in (17).

We have considered two types of algorithms for average consensus. One is based on a simple average computation, in which node i and each node $j \in N_i$, have the same weight in the convergence updating rule. Fig. 3 illustrates the evolution of the consensus algorithm for each node. As one can, the values slowly converge to the global average in an iterative way. Notice that the nodes affected by the isolated event, are rapidly suppressed by overall average related to the value with the highest incidence.

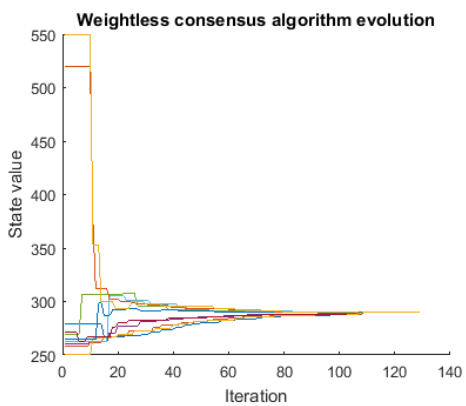
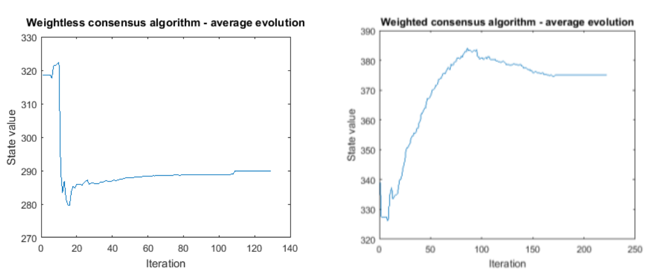


Fig. 3: Running average consensus on a 10-node random topology – compatible with Tmote Sky (CONTIKI OS)

The final convergence value does not reflect the occurrence of any deviations from the global average. The evolution of average value is illustrated in Fig. 4a. Final value is located around 290.



4(a): simple average consensus

4(b): weighted average consensus

Fig. 4: Evolution of average value.

The other evaluated solution is based on the proposed weighted average consensus. For comparison reasons the nodes were initiated with the same initial values, and the topology is also the same. The results are illustrated in Fig. 5.

As one can see, the final convergence value is different from the previous results. Notice that the initial values update in an iterative way and asymptotically converge to the global average. In this case, the deviated values have a higher influence over the updating steps, pulling up the neighbor values towards consensus. In this way the final value, which is higher than the initial global average, refer to the incidence of an event. The evolution of the global average is illustrated in Fig. 4b. Final value is around 375.

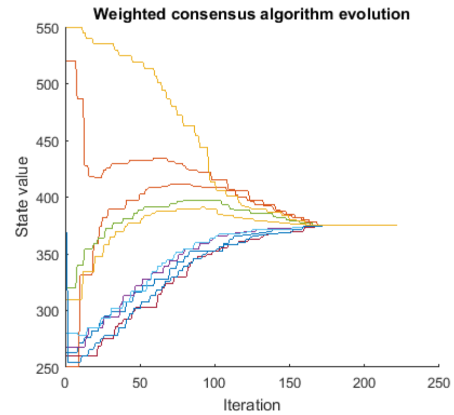


Fig. 5: Running weighted average consensus on a 10-node random topology – compatible with Tmote Sky (CONTIKI OS)

Although this method provides an improved event detection mechanism, one should consider the additional iterations that are necessary for the computation. In comparison to the first evaluation, this solution requires 30% more iterations.

V. CONCLUSION

The paper discussed a particular approach for average consensus algorithms, tailored for event detection. There are specific situations were estimating the global average value is not enough to detect the occurrence of an event. Most of the time, the incidence of isolated deviations is iteratively suppressed by overall average, related to the value with the highest incidence. This represents an important direction of study, mainly because detection of initial deviations in an early stage can bring important savings or benefits.

The proposed solution aims at correcting the convergence value using a weighted average consensus mechanism. A set of weight are locally assigned through in-neighbourhood data exchange based on an estimation of distances between neighbour nodes. Thus, the updating rule computes the current convergence value keeping a high priority for the closest neighbours. This ensures an asymptotically converge to the global average. The updating rule is based on a weighted average, with priority for the deviated measured values. In addition, the method uses a corrective factor which has been noted as strongly correlated to the dynamics of the convergence.

The weighted average consensus is developed mainly for improved event detection. The proposed algorithm only returns a better global average, in case of isolated events. To bring the most of it, one should consider the integration of artificial intelligence into the event detection system for smarter decision making. As the evaluation states, the proposed method requires more iterations to reach the final average value. The problem that emerged, referred to energetic reasons specific to embedded devices, is to select the optimal weights that maintain a cost-effective algorithm and to provide useful data, in the same time. This work is only a preliminary step in this research topic and captures the potential of weighted average consensus for improved event detection systems. Current and future work relates to defining a method for finding an optimal corrective factor. We also consider the integration of a mechanism for improving the convergence speed, similar to the Neighborhood algorithm.

Acknowledgment

*This work has been been funded by the University "Politehnica" of Bucharest, through the "Excellence Research Grants" Program, UPB-GEX. Identifier: UPB-EXCELENTA-2016 "New Dynamic Consensus Algorithms for Building Thermal Energy Management (CONSENSYS), Contract number 20/26.09.2016.

References

[1] P. Tiwari, V. P. Saxena, R. G. Mishra, D. Bhavsar, "Wireless Sensor Networks: Introduction, Advantages, Applications and Research Challenges," in *HCTL Open International Journal of Technology Innovations and Research (IJTIR)*, Vol. 14, pp. 1-11, Apr. 2015.

[2] K. Maraiya, K. Kant, N. Gupta, "Application based Study on Wireless Sensor Network," *International Journal of Computer Applications (0975-8887)*, Vol. 21-No.8, May 2011.

[3] A. Dobrin, G. Stamatescu, C. Dragana, V. Sgarciu, "Cloud Challenges for Networked Embedded Systems: a Review," in *20th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 866-871, Oct. 2016.

[4] C. Dragana, G. Stamatescu, L. Ichim, D. Popescu, "Interlinking Unmanned Aerial Vehicles with Wireless Sensor Networks for Improved Large Area Monitoring," in *The 4th International Conference on Control, Decision and Information Technologies (CoDIT'17)*, pp. 1-6, April 2017.

[5] F.L. Lewis, H. Zhang, K. Hengster-Movric, A. Das, "Cooperative Control of Multi-Agent Systems," Communications and Control Engineering, *Springer-Verlag London,* 2014.

[6] R. Olfati-Saber, J. A. Fax, R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," in *Proceedings of the IEEE*, Vol. 95, pp. 215-233, Mar. 2007.

[7] R. Olfati-Saber and R. M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and time-delays," in *IEEE Trans. on Automatic Control*, Vol. 49, pp. 1520-1533, 2004.

[8] P. Di Lorenzo, S. Barbarossa, "Distributed Estimation and Control of Algebraic Connectivity over Random Graphs," in *IEEE Transactions on Signal Processing*, Vol. 62, no. 21, pp. 5615-5628, Nov. 2014.

[9] A. Bertrand and M. Moonen, "Distributed computation of the Fiedler vector with application to topology inference in ad hoc networks," in *Signal Processing*, Vol. 93, no. 5, pp. 1106–1117, 2013.

[10] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, R. Sukthankar, "Decentralized estimation and control of graph connectivity in mobile sensor networks," in *Proc. American Control Conference*, pp. 2678-2683, Jun. 2008.

[11] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. ACM Symposium on Theory of Computing*, pp. 561–568, 2004.

[12] M. El Chamie, G. Neglia, K. Avrachenkov, "Reducing communication overhead for average consensus," in *IFIP Networking Conference*, pp. 1-8, 2013.

[13] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Systems and Control Letters*, Vol. 53, pp. 65-78, 2004.

[14] J. Tsitsiklis, "Problems in decentralized decision making and computations," Ph.D. dissertation, Lab. Information and Decision Systems, MIT, Cambridge, MA, 1984.

[15] K. Avrachenkov, M. El Chamie, G. Neglia, "A local average consensus algorithm for wireless sensor networks," in *DCOSS*, pp. 1–6, 2011.

[16] Y. Chen, R. Tron, A. Terzis, R. Vidal, "Corrective consensus: converging to the Exact Average," in *49th IEEE Conference on Decision and Control*, pp. 1-8, 2010.

[17] G. Jing, Y. Zheng, L. Wang, "Consensus of multi-agent systems with distance-dependent communication," in *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1-15, Aug. 2016.