Graph Laplacian based Matrix Design for Finite-Time Distributed Average Consensus

Alain Kibangou

Graph Laplacian based Matrix Design for Finite-Time Distributed Average Consensus

Alain Y. Kibangou

***Abstract*— In this paper, we consider the problem of finding a linear iteration scheme that yields distributed average consensus in a finite number of steps D. By modeling interactions between the nodes in the network by means of a time-invariant undirected graph, the problem is solved by deriving a set of D Laplacian based consensus matrices. We show that the number of steps is given by the number of nonzero distinct eigenvalues of the graph Laplacian matrix. Moreover the inverse of these eigenvalues constitute the step-sizes of the involved Laplacian based consensus matrices. When communications are made through an additive white Gaussian noise channel, based on an ensemble averaging method, we show how average consensus can be asymptotically reached. Performance analysis of the suggested protocol is given along with comparisons with other methods in the literature.**

I. Introduction

Recently, a great effort was made by both control and signal processing communities to provide coordination and estimation algorithms for static and mobile wireless sensor networks [1], [2], [3]. Such an effort is particularly due to the great potential impact of mobile wireless sensor networks (Mobile devices with sensing, computing and communication capabilities) to numerous applications including surveillance of hazardous plants, and more others. Consensus plays a key role in most of the distributed algorithms proposed in the literature. The fundamental aim of consensus is that the nodes in the network reach an agreement on a common value by negotiating with their neighbors. In particular, in this paper, we focus on average consensus where the common value consists in the average of the initial ones.

Average consensus can be reached using a linear iterations scheme where each node repeatedly updates its value as a weighted linear combination of its own value and those of its neighbors. Therefore, at each time-step, each node only has to transmit a single value to each of its neighbors. Based on such a scheme, several algorithms have been proposed in the literature. However, in the most of the proposed algorithms, the weights are chosen so that all the nodes asymptotically converge to the same value.

A key feature of a mobile wireless sensor network is related to the network topology. It is common to represent the nodes in the network as vertices of a given graph where edges represent the existence of a communication between them. This powerful modeling tool allows using spectral graph theory for analyzing the associated algorithms and protocols. Indeed, several features of a network can be characterized by means of spectral properties of the so-called graph Laplacian. For example, the second smallest eigenvalue of the graph Laplacian matrix is known to have the main role in the convergence time of various distributed algorithms [4]. Therefore, computing the eigenvalues of the Laplacian matrix is an important question, generally solved through standard matrix tools that lead to centralized scheme. Recently, two interesting attempts to decentralize the eigendecomposition of a Laplacian matrix have been introduced in the literature. The first one is that of Fransceschelli and his co-workers [5] where the basic idea is to make the nodes states oscillate only at frequencies corresponding to the eigenvalues of the network topology. The problem is then mapped into a signal processing one that each agent can efficiently and independently solve by applying the Fast Fourrier Transform (FFT). The second work is that of Sahai and co-workers [6] where a similar concept based on the discretization of the wave equation was used.

Motivated by these results, we investigate the use of Laplacian eigenvalues for designing a sequence of weight matrices that allow achieving consensus in finite-time.

A number of authors have studied finite-time consensus in the framework of discrete-time systems. In [7], finite-time average consensus was briefly discussed. However, the described method requires the graph to be fully connected for at least one time-step. A data aggregation based algorithm was proposed in [8]. Such an approach requires additional memory compared to linear iterations strategies. Another approach requiring less memory, but linear iterations, is derived in [9] by calculating weights so that the observability matrix of the network is contained in some defined space. The basic idea is that given enough time, the nodes will have observed enough to reconstruct the initial state of the system. At which time, they can compute the correct average. In [10], the same authors have shown that each node can calculate the consensus value as a linear combination of its own past values over at most D time-steps, D being the degree of the minimal polynomial of the associated weight matrix. The computation of a given matrix rank and nullspace is required. Knowing the computation cost of such tasks, these approaches necessitates consequent computation capabilities. In [11], based on properties of de Bruijn's graph and block Kronecker product, it has been shown that the average consensus problem can be reached in finite time if the number of nodes is an exact power of the out-degree of the communication graph. Another interesting contribution is that in [12] where finite-time average consensus problem

This work was partly supported by the EU project FeedNetBack

Alain Y. Kibangou is with GIPSA-Lab, University Joseph Fourier, CNRS, 11 rue des Mathématiques, BP 46, F-38402 Grenoble Cedex, France. alain.kibangou@ujf-grenoble.fr

is formulated as a matrix factorization problem. However, the proposed approach is fully centralized and requires scheduling of nodes connection. Recently, for time-invariant topologies and in the perfect information exchange case, i.e. without channel noise nor quantization, we have shown that the finite-time average consensus problem can be solved as a matrix factorization problem with joint diagonalizable matrices [13]. By following the same concept, in this paper, we synthesize a new family of finite-time consensus matrices. In fact, we show that adequately selected consensus matrices based on Laplacian eigenvalues allow achieving average consensus in a finite number of steps.

Then, by adequately and periodically restarting the finite-time consensus algorithm, we show that, in the noisy case, average consensus can be achieved asymptotically with desirable asymptotic properties.

II. Problem setting

Let us consider a network modeled as an undirected connected graph $\mathscr{G} = (\mathscr{X}, \mathscr{E})$ consisting of a set of nodes $\mathscr{X} = \{x_1, \cdots, x_N\}$ and a set of edges $\mathscr{E} \subseteq \mathscr{X} \times \mathscr{X}$. We denote by $\mathscr{N}_i$ the set of nodes that can transmit information to node x_i. Its cardinality is denoted N_i. The graph Laplacian $\mathbf{L}$ is defined as the matrix with entries l_{ij} given by:

$$l_{ij} = \begin{cases} N_i & \text{if } i = j \\ -1 & \text{if } j \in \mathscr{N}_i \\ 0 & \text{elsewhere} \end{cases}.$$

Since the graph is undirected, the graph Laplacian is symmetric ($\mathbf{L} = \mathbf{L}^T$). Its eigenvalues, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, contain very significant information about the topology of the graph $\mathscr{G}$. In particular, we have $\lambda_1 = 0$ that admits $\mathbf{1}_N$, an N-length all ones column-vector, as eigenvector.

Now, let us assume that each node n holds an initial scalar value $x_n(0) \in \Re$ and $\mathbf{x}(0) = \begin{pmatrix} x_1(0) & \cdots & x_N(0) \end{pmatrix}^T$ denotes the vector of the initial values on the network. We are interested in computing the average of the initial values by means of a distributed algorithm, in which the nodes only communicate with their neighbors. A common way to achieve such task is to resort to linear iterations schemes where each node repeatedly updates its value as a linear combination of its own value and that of its neighbors:

$$x_n(k+1) = w_{nn} x_n(k) + \sum_{m \in \mathscr{N}_n} w_{nm} x_m(k), \tag{1}$$

or in matrix form:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k), \tag{2}$$

where the off-diagonal entries $w_{i,j}$ of $\mathbf{W}$ are nonzero if and only if $x_j \in \mathscr{N}_i$. The average consensus is reached if

$$\lim_{k \to \infty} \mathbf{x}(k) = \mathbf{J}_N \mathbf{x}(0), \quad \text{with} \quad \mathbf{J}_N = \frac{1}{N} \mathbf{1}_N \mathbf{1}_N^T.$$

The statement above means:

$$\lim_{k \to \infty} \mathbf{W}^k = \mathbf{J}_N.$$

It is now well known that consensus is achieved if and only if $\mathbf{W}$ admits 1 as a simple eigenvalue while the remaining eigenvalues have magnitude strictly less than 1, the left and right eigenvectors of $\mathbf{W}$ associated with the eigenvalue 1 being $\frac{1}{\sqrt{N}}\mathbf{1}$. With these conditions asymptotic convergence is guaranteed. That is the case for the Laplacian based consensus matrix $\mathbf{W} = \mathbf{I} - \gamma\mathbf{L}$, with $0 < \gamma < 1/N_{max}$, $N_{max} = \max\{N_1, \cdots, N_N\}$ [14].

Speed of convergence of consensus algorithm is generally governed by the second largest eigenvalue of $\mathbf{W}$. Therefore, several authors have proposed different methods to speed up the convergence of consensus algorithms [4], [15], [16]. Although fast convergence is sufficient in numerous cases, it is sometimes desirable to achieve convergence to the exact value in a finite number of steps.

In [13], the finite-time consensus problem was formulated as a matrix factorization problem: Find a set of matrices $\{\mathbf{W}_i\}_{i=1,\cdots,D}$, consistent with the network topology, so that

$$\prod_{i=1}^{D} \mathbf{W}_i = \mathbf{J}_N. \tag{3}$$

Recall that a matrix is said to be consistent with the network topology if its off-diagonal (i, j)th entry equals to zero for $j \notin \mathscr{N}_i$.

In [13], a family of matrices parameterized as $\mathbf{W}_i = (\alpha_i + N_{max}\beta)\mathbf{I} - \beta\mathbf{L}$ was synthesized. These matrices do not fulfill the conservation property; they are not left stochastic. In this paper, our goal is to derive a set of Laplacian based consensus matrices, $\mathbf{W}_i = \mathbf{I} - \alpha_i\mathbf{L}$ so that (3) is fulfilled. It is equivalent to design a sequence of step-sizes α_i so that exact average consensus is achieved in finite-time.

III. Laplacian based consensus matrices step-size sequence design for Finite-Time average consensus

The Laplacian matrix $\mathbf{L}$ being symmetric, its diagonalization yields

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \ \mathbf{U}\mathbf{U}^T = \mathbf{I},$$

where $\mathbf{\Lambda} = diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$ and $\mathbf{U} = \begin{pmatrix} \frac{1}{\sqrt{N}}\mathbf{1} & \tilde{\mathbf{U}} \end{pmatrix}$ with $\tilde{\mathbf{U}}^T\tilde{\mathbf{U}} = \mathbf{I}_{N-1}$ and $\tilde{\mathbf{U}}^T\mathbf{1} = \mathbf{0}$.

As a consequence, the consensus matrices $\mathbf{W}_i$ can be written according to the eigenvectors and eigenvalues of the Laplacian matrix as follows:

$$\mathbf{W}_i = \mathbf{U}\left(\mathbf{I} - \alpha_i\mathbf{\Lambda}\right)\mathbf{U}^T.$$

We can now rewrite equation (3) as:

$$\mathbf{U}\left(\prod_{i=1}^{D}\left(\mathbf{I} - \alpha_i\mathbf{\Lambda}\right)\right)\mathbf{U}^T = \mathbf{J}_N, \tag{4}$$

or equivalently:

$$\mathbf{U}\left(\prod_{i=1}^{D}\left(\mathbf{I} - \alpha_i\mathbf{\Lambda}\right)\right)\mathbf{U}^T = \mathbf{U}diag(1 \quad 0 \cdots 0)\mathbf{U}^T. \tag{5}$$

We can now state the following theorem:

Theorem 1: Let $\lambda_2 \neq \lambda_3 \neq \cdots \neq \lambda_{D+1} \neq 0$ be the D distinct nonzero eigenvalues of the graph Laplacian matrix $\mathbf{L}$, then

the set of consensus matrices $\mathbf{W}_i = \mathbf{I} - \frac{1}{\lambda_{i+1}}\mathbf{L}$, $i = 1, \cdots, D$, allows reaching average consensus in D steps.

Proof: From equation (5) we get:

$$\prod_{i=1}^{D}(1-\alpha_i\lambda_j) = \begin{cases} 1 \text{ for } j=1 \\ 0 \text{ elsewhere} \end{cases}.$$

Since $\lambda_1 = 0$, we have to solve

$$\prod_{i=1}^{D}(1-\alpha_i\lambda_j) = 0, \ j = 2,3,\cdots,N.$$

By taking into account multiplicities of the eigenvalues, we have only D distincts equations. A solution of the resulting system of equations is then given by setting each α_i equal to the inverse of a given Laplacian eigenvalue. ■

Owing to the above theorem, we can build $D!$ different sequences. Both give rise to the exact average consensus after D steps but the transient behavior could be very different. Selecting the best sequence is out of the scope of this paper. However, we will emphasize the impact of such a choice in the next section by considering noisy data exchanges.

Example 1: Consider the undirected graph depicted in Fig. 1[4]. The weights on the nodes and edges indicate the optimal weights to maximize the rate of convergence for the system, as provided by the method in [4]. The associated

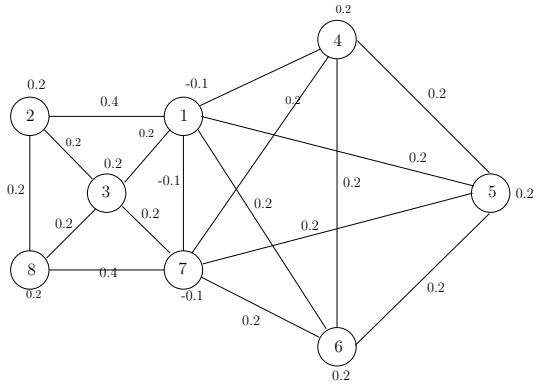

Fig. 1. Graph with optimal weights according to [4].

graph Laplacian has the following distinct nonzero eigenvalues $\{1.5226,\ 3.6972,\ 4.5519,\ 5.0000,\ 6.9254,\ 7.3028\}$. Therefore exact average consensus can be achieved in $D = 6$ steps. Let us consider two sequences of consensus matrices. In the first one the distinct nonzero eigenvalues are placed in ascending order while in the second they are in descending order. As depicted in Fig. 2, even the optimal fast consensus algorithm needs several iterations to give accurate results. As explained before, we can also note that the two sequences of consensus matrices give rise to different transient behavior (see also Fig. 3 depicting the nodes trajectories.

IV. Analysis in non-ideal cases

The results derived in the previous section allow reaching the exact average consensus in a finite number of steps when perfect data exchange is considered. In this section, we first

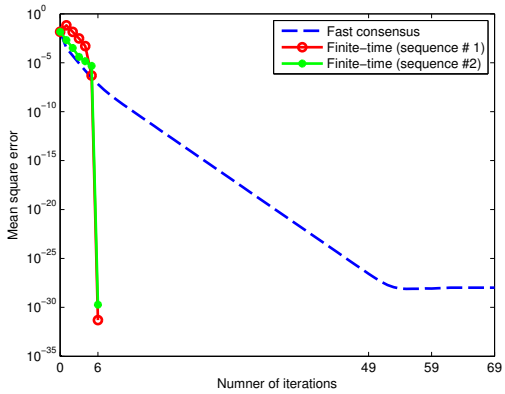

Fig. 2. Comparison between finite-time consensus and fast consensus as proposed in [4].

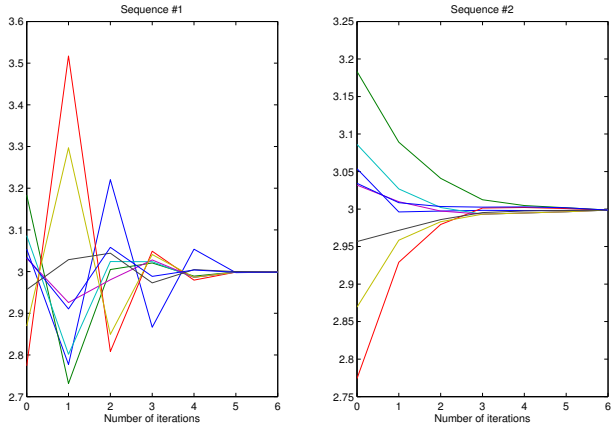

Fig. 3. Nodes trajectories obtained with two sequences of Laplacian based consensus matrices.

study the effect of errors on Laplacian eigenvectors and then the effect of additive noise on average consensus.

A. Effect of estimation errors on Laplacian eigenvalues

Let us assume that the distinct nonzero eigenvalues λ_{i+1}, $i = 1, \cdots, D$, are known up to an error ε_{i+1}. The consensus matrices are then built with erroneous eigenvalues $\tilde{\lambda}_{i+1}$:

$$\mathbf{W}_i = \mathbf{I} - \frac{1}{\tilde{\lambda}_{i+1}}\mathbf{L}.$$

Due to the errors on the eigenvalues the product of these consensus matrices is no longer equals to $\mathbf{J}_N$. In the sequel, we quantify the corresponding square error.

The product of the consensus matrices can be written as:

$$\prod_{i=1}^{D}\mathbf{W}_i = \mathbf{U}\mathbf{\Delta}\mathbf{U}^T, \tag{6}$$

where $\mathbf{\Delta} = diag\begin{pmatrix} 1 & \Delta_2\mathbf{1}_{m_2} & \cdots & \Delta_{D+1}\mathbf{1}_{m_{D+1}} \end{pmatrix}$ with m_j the multiplicity of λ_j, and $\Delta_j = -\frac{\varepsilon_j}{\tilde{\lambda}_j}\prod_{i=1, i\neq j}^{D}\left(1 - \frac{\lambda_j}{\tilde{\lambda}_{i+1}}\right)$. Equivalently, we can rewrite (6) as:

$$\prod_{i=1}^{D}\mathbf{W}_i = \mathbf{J}_N + \tilde{\mathbf{J}} = \mathbf{J}_N + \mathbf{U}\bar{\mathbf{\Delta}}\mathbf{U}^{\mathbf{T}},$$

where $\overline{\mathbf{\Delta}}$ is obtained by setting the first diagonal entry of $\mathbf{\Delta}$ equal to zero. Therefore, after D iterations the nodes have computed $\mathbf{x}(D) = \bar{\mathbf{x}} + \tilde{\mathbf{J}}\mathbf{x}(0)$, $\bar{\mathbf{x}}$ denoting the exact average of the initial values. The computation of the square error yields:

$$e = (\mathbf{x}(D) - \bar{\mathbf{x}})^T (\mathbf{x}(D) - \bar{\mathbf{x}}) = \left\|\tilde{\mathbf{J}}\mathbf{x}(0)\right\|_F^2 \leq \left\|\tilde{\mathbf{J}}\right\|_F^2 \|\mathbf{x}(0)\|^2,$$

where $\|.\|_F$ denotes the Frobenius norm. Equivalently, we get:

$$e \leq \sum_{j=2}^{D+1} m_j \frac{\varepsilon_j^2}{\tilde{\lambda}_j^2} \prod_{i=1, i\neq j} \left(1 - \frac{\lambda_j}{\tilde{\lambda}_{i+1}}\right)^2 \|\mathbf{x}(0)\|^2. \quad (7)$$

This upper bound shows that the square error e can be arbitrarily small if the relative estimation errors associated with the eigenvalues with greater magnitude are sufficiently small. However, this result point out the fact that estimators of the Laplacian eigenvalues should be particularly efficient.

B. Effect of additive noise

Now, let us consider the case of noisy data exchange:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) + \mathbf{v}(k+1),$$

where $\mathbf{v}$ denotes the additive noise.

In such a case, standard average consensus algorithms are known to suffer from a linear increase of noise variance with time [17]. In order to limit or circumvent this undesired effect, alternative methods for designing the weights of average consensus algorithms have been proposed in the literature. For example, in [17], it was shown how to derive the weighting matrix in order to minimize the deviation between the nodes values. Starting from a Perron matrix, decreasing step-size methods have been proposed in the literature (see [18], [19] and references therein). A first order difference equation approach, which is resilient to noise, has been also proposed in [20]. In [13], the author proposed a sequence averaging approach. Herein we give an analysis of such approach while considering a more general averaging approach.

The main idea of the suggested protocol is to run periodically the finite-time average consensus algorithm. The results of each run are then averaged in order to compensate for the deviations induced by the additive noise. To do so, the finite-time algorithm is restarted with $\mathbf{x}(0)$ with appropriately chosen weight c_n:

$$\begin{aligned}\mathbf{x}(k+(n-1)D) &= (1-\delta_{k1})\mathbf{W}_k\mathbf{x}(k-1+(n-1)D) \\ &\quad +\delta_{k1}c_n\mathbf{W}_k\mathbf{x}(0), \end{aligned} \quad (8)$$

where $k = 1, \cdots, D$, $n = 1, \cdots, \infty$, δ_{ij} denotes the Kronecker delta, i.e. $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ elsewhere.

After D steps of the finite-time average consensus algorithm, we get $\mathbf{x}(nD)$ and we can compute the weighted averaged value

$$\mathbf{y}(nD) = C(n) \sum_{m=1}^{n} \mathbf{x}(mD), \quad \text{with} \quad C(n) = \left(\sum_{m=1}^{n} c_m\right)^{-1}. \quad (9)$$

In this protocol, each node keeps in memory its initial value and re-send its scaled version periodically to its neighbors. With a period equals to D, in the noiseless case, one can note that

$$\mathbf{x}(nD) = \prod_{i=D}^{i} \mathbf{W}_i c_n \mathbf{x}(0) = c_n\bar{\mathbf{x}} = \frac{c_n}{N}\mathbf{J}_N\mathbf{x}(0)$$

that yields $\mathbf{y}(nD) = \bar{\mathbf{x}}$. However, when the data exchange is noisy, we get

$$\mathbf{x}(nD) = c_n\bar{\mathbf{x}} + \sum_{i=0}^{D-2}\prod_{j=0}^{i}\mathbf{W}_{D-j}\mathbf{v}(nD-i-1) + v(nD) \quad (10)$$

Theorem 2: Considering the consensus protocol (8) with sequence averaging (9), if the noise sequence $\{\mathbf{v}(k+(n-1)D)\}$ is independent with zero mean and covariance $\sigma^2\mathbf{I}_N$, then after n runs, the mean square error $e(n) = \mathbb{E}\left[(\mathbf{y}(nD) - \bar{\mathbf{x}})^T (\mathbf{y}(nD) - \bar{\mathbf{x}})\right]$ on the exact consensus value, is given by:

$$e(n) = nNC(n)^2\sigma^2\left(1 + \frac{1}{N}\sum_{k=1}^{N}\sum_{i=0}^{D-2}\prod_{j=0}^{i}\left(1 - \frac{\lambda_k}{\lambda_{D-j+1}}\right)^2\right), \quad (11)$$

with $\lambda_1, \cdots, \lambda_N$ the graph Laplacian eigenvalues, the $D+1$ first being distinct and $\lambda_1 = 0$, $\mathbb{E}[.]$ denoting the mathematical expectation.

Proof: Let us defined $\mathbf{z}(nD) = \mathbf{y}(nD) - \bar{\mathbf{x}}$. Using (10), we get:

$$\mathbf{z}(nD) = C(n)\sum_{m=1}^{n}\left(\sum_{i=0}^{D-2}\prod_{j=0}^{i}\mathbf{W}_{D-j}\mathbf{v}(mD-i-1) + v(mD)\right).$$

We have then to compute $e(n) = \mathbb{E}\left[\mathbf{z}^T(nD)\mathbf{z}(nD)\right] = trace(\mathbb{E}\left[\mathbf{z}(nD)\mathbf{z}^T(nD)\right])$. Owing to the independence property, the matrix $\mathbb{E}\left[\mathbf{z}(nD)\mathbf{z}^T(nD)\right]$ can be written as:

$$\mathbb{E}\left[\mathbf{z}(nD)\mathbf{z}^T(nD)\right] = nC(n)^2\sigma^2(\mathbf{Q} + \mathbf{I}_N),$$

where $\mathbf{Q} = \sum_{i=0}^{D-2}\prod_{j=0}^{i}\mathbf{W}_{D-j}\prod_{j=i}^{0}\mathbf{W}_{D-j}^T$. Hence

$$e(n) = nNC(n)^2\sigma^2(1 + trace(\mathbf{Q})/N).$$

Let us now compute $trace(\mathbf{Q})$ to end the proof. Knowing that $\mathbf{W}_{D-j} = \mathbf{I} - \frac{1}{\lambda_{D-j+1}}\mathbf{L}$ and using the eigendecomposition of $\mathbf{L}$, we can rewrite the consensus matrices as $\mathbf{W}_{D-j} = \mathbf{U}\left(\mathbf{I} - \frac{1}{\lambda_{D-j+1}}\mathbf{\Lambda}\right)\mathbf{U}^T$. As a consequence, we get:

$$\mathbf{Q} = \sum_{i=0}^{D-2}\mathbf{U}\prod_{j=0}^{i}\left(\mathbf{I} - \frac{1}{\lambda_{D-j+1}}\mathbf{\Lambda}\right)^2\mathbf{U}^T.$$

Hence:

$$\begin{aligned} trace(\mathbf{Q}) &= \sum_{i=0}^{D-2} trace\left(\prod_{j=0}^{i}\left(\mathbf{I} - \frac{1}{\lambda_{D-j+1}}\mathbf{\Lambda}\right)\right) \\ &= \sum_{k=1}^{N}\sum_{i=0}^{D-2}\prod_{j=0}^{i}\left(1 - \frac{\lambda_k}{\lambda_{D-j+1}}\right)^2 \end{aligned}$$

■

Based on this theorem, we can note that by appropriately selecting the weighting coefficients c_n, the mean square error can be decreased faster than in the sequence averaging approach proposed in [13], which is a particular case of the approach proposed herein. However, as pointed out in section 2, in (11), the effect of the distinct eigenvalues used for designing the consensus matrices is not the same. The ordering of these eigenvalues could have a significant impact on the mean square error.

Example 2: Let us consider a network with 6 nodes. The associated topology is a circle one. The nonzero graph Laplacian eigenvalues are 1, 3, and 4. The sequence of consensus matrices obtained with the eigenvalues ordered as $\{4,3,1\}$ gives rise to a matrix $\mathbf{Q}$ with higher trace ($trace(\mathbf{Q}) = 20$) whereas for the second sequence, obtained with the eigenvalues ordered as $\{3,4,1\}$, we get the lower value of the trace : $trace(\mathbf{Q}) = 3.5$. As a consequence we get best results, in the theoretical mean square error sense, for the second sequence (see Fig. 4).

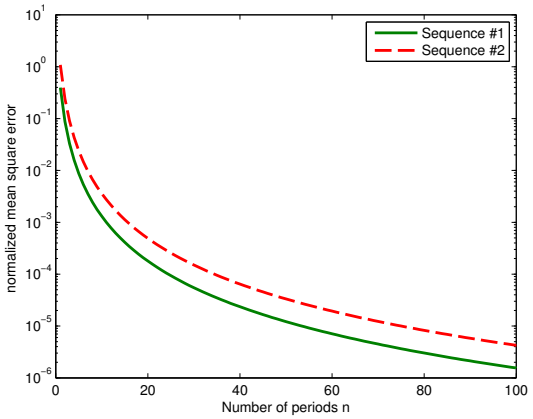

Fig. 4. Theoretical normalized mean square error (case of a circle with 6 nodes with the sequence averaging policy $c_n = n$).

Based on Theorem 2, the asymptotic behavior of the proposed method is stated in the following corollary.

Corollary 1: Considering the consensus protocol (8)-(9), suppose that the noise sequence $\{\mathbf{v}(.)\}$ is independent with zero mean and covariance $\sigma^2\mathbf{I}_N$. Then the sequence $\{\mathbf{y}(nD)\}$ asymptotically converges to the average value $\bar{\mathbf{x}}$ of the initial state:

$$\lim_{n\to\infty} e(n) = 0.$$

V. Simulation results

In this section, we consider a network of $N = 27$ nodes whose communications are modeled by means of a cubic lattice graph. The nodes communicates through an additive white Gaussian noise channel with variance 10^{-4}. The performance are evaluated by means of the Normalized Mean Square Error (NMSE) defined as $NMSE_k = \|\mathbf{x}(k) - \bar{\mathbf{x}}\|^2 / \|\bar{\mathbf{x}}\|^2$, $\bar{\mathbf{x}}$ being the average of the initial values of the nodes. The results below are averaged values over 100 independent Monte Carlo runs.

The cubic lattice graph is defined to be a graph $\mathscr{G}$, whose vertices are the ordered triplets on n symbols, such that two vertices are adjacent if and only if they have two coordinates in common [21]. For a graph with n^3 vertices, the graph Laplacian matrix has only four distinct eigenvalues: 0,n,2n, and 3n. Therefore, the set of Laplacian based consensus matrices allowing to reach consensus in $D = 3$ steps is given by: $\mathbf{W}_1 = \mathbf{I} - \frac{1}{n}\mathbf{L}$, $\mathbf{W}_2 = \mathbf{I} - \frac{1}{2n}\mathbf{L}$, and $\mathbf{W}_3 = \mathbf{I} - \frac{1}{3n}\mathbf{L}$, with $n = 3$ for the simulated case.

First of all, we compare the theoretical NMSE with that obtained by simulating an average consensus problem through the network. From 5, we can note that the obtained NMSE is close to the theoretical ones.

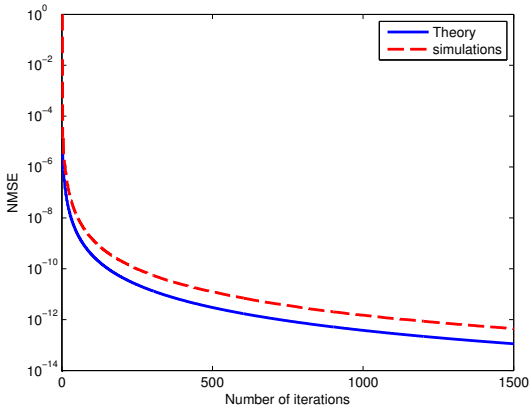

Fig. 5. Comparison of the experimental NMSE with theory for the sequence averaging policy $c_n = n$.

Now, we give the results of the comparison of the proposed scheme with the sequence averaging scheme proposed in [13] and with a first order difference based algorithm. Recall that the first order difference scheme is such that

$$\begin{aligned} \mathbf{\Phi}(k+1) &= \mathbf{x}(0) + (\mathbf{I} - \gamma\mathbf{L})\mathbf{\Phi}(k) \\ \mathbf{x}(k) &= \mathbf{\Phi}(k) - \mathbf{\Phi}(k-1) \end{aligned}$$

Its noise resilience comes from the difference operation which counteracts the effect of the eigenvalue 1 of the weight matrix. We set $\gamma = 1/2N_{max}$.

Note that the comparison with [13] is based on the sequence averaging process not in the derived finite-time consensus matrices.

Figure 6 depicts the NMSE obtained with the three compared methods. We can note that the proposed method outperforms the two others in terms of precision. Moreover even after the first $D = 3$ iterations the finite-time algorithms are more precise than the asymptotic based one.

In figure 7 where the nodes trajectory are depicted, we can see that after the D first iterations the local values are very close to the actual average value. We note that the first order difference method allows reaching consensus in a given bound around the actual value, while exact consensus is achieved with the proposed scheme. The precision is enough to assume that all the nodes reach the same value.

VI. Conclusion

Graph Laplacian is a very useful tool for studying the behavior of consensus algorithms. In particular, it has been

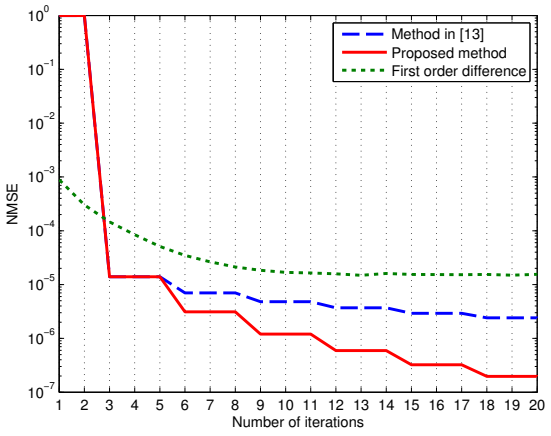

Fig. 6. Zoom of the evaluation of the NMSE for a cubic lattice graph with 27 nodes.

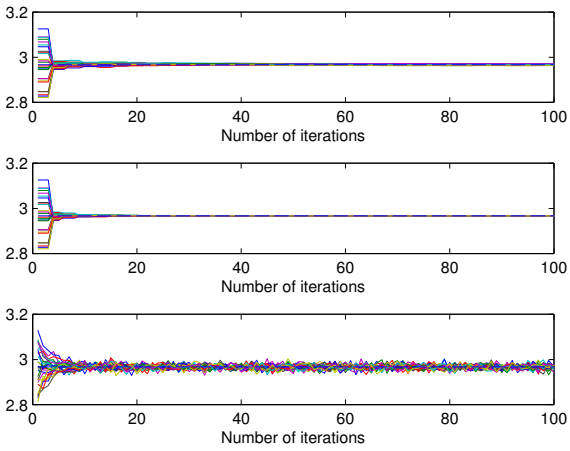

Fig. 7. Nodes trajectory for the compared methods in the case of a cubic lattice graph.

used for characterizing speed of convergence of Laplacian based consensus algorithms. It has been shown that under some conditions on the Laplacian based consensus matrix, consensus can be achieved asymptotically. In this paper, we have shown that by adequately varying the step-size of the Laplacian based consensus matrix, consensus can be achieved in a finite number of steps given by the number of nonzero distinct eigenvalues of the graph Laplacian matrix. Moreover, the sequence of step-sizes is given by the inverse of the Laplacian eigenvalues. At first glance, the implementation of this method requires a global knowledge of the network. It is then well suited for networks configured by an operator having such knowledge. Fully distributed approaches can be derived by distributively estimating the eigenvalues of the Laplacian matrix. Hopefully some solutions have been recently reported in the literature [5], [6]. Our method can then make use of such results. Since the estimation of the eigenvalues can be more or less imprecise, we have evaluate the effect of erroneous eigenvalues on finite-time average consensus. We have also analyzed the proposed method when communications are made through an additive white Gaussian noise. We have proposed a promising sequence averaging method leading to an asymptotic convergence towards the exact average value.

REFERENCES

[1] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proc. of the joint 44th IEEE Conf. on Decision and Control (CDC) and European Control Conf (ECC)*, pages 2996–3000, Seville, Spain, December 2005.

[2] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49:1520–1533, 2004.

[3] S. Barbarossa and G. Scutari. Bio-inspired sensor network design. *IEEE Signal Proc. Magazine*, 24:26–35, May 2007.

[4] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems Control Lett.*, 53:65–78, 2004.

[5] M. Franceschelli, A. Gasparri, A. Giua, and C. Seatzu. Decentralized Laplacian eigenvalues estimation for networked multi-agent systems. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 2717–2722, Shanghai, P.R. China, 2009.

[6] T. Sahai, A. Speranzon, and A. Banaszuk. Wave equation based algorithm for distributed eigenvector computation. In *49th IEEE Conference on Decision and Control*, pages 7308–7315, Atlanta, GA, USA, 2010.

[7] D.B. Kingston and R.W. Beard. Discrete-time average consensus under switching network topologies. In *Proc. of American Control Conference (ACC)*, Minneapolis, Minenesota, USA, 2006.

[8] N. Lechevin, C.A. Rabbath, and Y. Zhang. Information broadcasting algorithm for finite-time reaching-at-risk consensus with application to weapon-target assignment. In *Proc. of American Control Conference (ACC)*, pages 3286–3291, St. Louis, MO, USA, June 10-12 2009.

[9] S. Sundaram and C. Hadjicostis. Distributed function calculation and consensus using linear iterative strategies. *IEEE Journal on Selected Areas in Communications*, 26(4):650–660, May 2008.

[10] S. Sundaram and C. Hadjicostis. Finite-time distributed consensus in graphs with time-invariant topologies. In *Proc. of American Control Conference (ACC)*, New York City, USA, July 11-13 2007.

[11] J.-C. Delvenne, R. Carli, and S. Zampieri. Optimal strategies in the average consensus problem. *Systems and Control Letters*, 58:759–765, 2009.

[12] C.-K. Ko. *On matrix factorization and scheduling for finite-time average consensus*. PhD thesis, California Institue of Technology, Pasadena, California, USA, January 2010.

[13] A.Y. Kibangou. Finite-time average consensus based protocol for distributed estimation over awgn channels. In *Proc. of the 50th IEEE Conference on Decision and Control (CDC)*, Orlando, Fl, USA, 2011.

[14] R. Olfati-Saber, A. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proc. of the IEEE*, 95(1):215–233, January 2007.

[15] E. Kokiopoulou and P. Frossard. Polynomial filtering for fast convergence in distributed consensus. *IEEE Trans. on Signal Processing*, 57:342–354, January 2009.

[16] E. Montijano, J.I. Montijano, and C. Sagues. Fast distributed consensus with Chebyshev polynomials. In *2011 American Control Conference*, pages 5450–5455, San Francisco, CA, USA, 2011.

[17] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67:33–46, 2007.

[18] C. Mosquera, R. Lopez-Valcarce, and S.K. Jayaweera. Distributed estimation with noisy exchanges. In *Proc. of SPAWC*, pages 236–240, Recife, Brazil, July 2008.

[19] S. Kar and J.M.F. Moura. Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Trans. on Signal Proc.*, 57(1):355–369, 2008.

[20] S. Barbarossa and G. Scutari. Decentralized maximum likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems. *IEEE Trans. on Signal Proc.*, 55(7):3456–3470, 2007.

[21] R. Laskar. Eigenvalues of the adjacency matrix of cubic lattice graphs. *Pacific Journal of Mathematics*, 29(3):623–629, 1969.