Optimizing Convergence Speed in Adaptive Consensus Algorithms

Nemanja Ilić, Miloš S. Stanković and Srdjan S. Stanković

***Abstract*—In this paper multi-step adaptive consensus algorithms for distributed information processing via networks of intelligent sensors are considered. Network nodes are assumed to have limited resources in view of the limited connectivity range and the local availability of measurements (limited sensing range). Consensus schemes aimed at achieving the weighted average of node states (local information processing results) by utilizing the locally available communication channels are discussed. The setting where the adaptive asymptotic consensus weights reflect assertions about the individual node quality as well as the real-time measurement availability is adopted. Within this context, the paper proposes practically convenient procedure for designing the adaptive communication weights exhibiting the fastest convergence speed of the resulting consensus scheme, and proves the claimed optimality. This issue is of great importance for reducing sensor network communication requirements and increasing its energy efficiency. A number of numerical examples are given, illustrating the algorithm behind the proposed procedure and demonstrating its effectiveness.**

***Index Terms*—Sensor networks, Adaptive consensus, Convergence speed, Optimization.**

I. INTRODUCTION

The problem of signal and information processing via sensor networks has attracted a great deal of scientific interest over the past few decades, *e.g.*, [1], [2]. Distributed algorithms have been in the focus of many researchers, due to their high potential for increasing the robustness and fault tolerance of the resulting schemes. Among various types of distributed schemes, consensus algorithms have emerged as one of the dominantly used protocols for reaching the agreement of different nodes in the network regarding variables connected to the global processing task [3], [4].

The application domain of consensus algorithms has been very wide. Consensus has been used in distributed state estimation [2], [5], [6], [7], fault diagnosis [8], [9], change detection [10], [11], [12], [13], optimization [14], target tracking [15], [16], [17], [18], [19], [20], etc. All of these solutions inherently adopt the local connectivity assumption, *i.e.*, nodes in the network are connected only to their neighboring nodes; all-to-all connections are usually not considered since they correspond to the centralized schemes.

One of the major challenges for the successful application of consensus schemes has been to appropriately address the problem of some sensors not receiving measurements connected to the global processing task at a given time instant, *i.e.*, the problem of local observability [21], [22]. This represents the special case of a more general setting with discrepant local processing results - different nodes in the network obtain local variables with different "quality", which can be related to both *a priori* assertions about the nodes (*e.g.*, connected to local measurement error covariances [9], [12], [13]), as well as to real-time assertions (*e.g.*, reflecting the fact that a node does or does not receive measurements at a given time instant [15], [17], or receives them with high or low probability [16], [23]). In any case, this issue has to be taken into account when designing the consensus communication scheme, so that the achieved agreement is dominantly influenced by the "high quality" nodes. The so-called adaptive consensus schemes [15], [16] have demonstrated their effectiveness in solving the problems with discrepant local processing results. The proposed algorithms in [23], [17] solve the problem by using multiple consensus iterations at each time instant and by adopting the desirable asymptotic behavior of the consensus scheme, resembling the two parallel passes scheme from [24].

When applying any communication procedure in the context of sensor networks, and, in particular, when applying the multi-step consensus schemes, it is very important to reduce the needed communication burden. This represents an issue that must be taken into account, having in mind the imposed energy efficiency imperatives when working with sensor networks. In this paper we address the problem of designing the adaptive consensus weights exhibiting the fastest convergence speed, and propose a practically convenient procedure as a solution. We also prove the optimality of the proposed solution, extending the classical results from [25]. The adopted problem setting is general and can be applied to different application scenarios, such as distributed state estimation, target tracking, change detection, optimization, etc. A number of numerical examples are given, illustrating the underlying steps behind the proposed procedure and demonstrating its effectiveness.

The outline of the paper is as follows. Problem setting and the used adaptive multi-step consensus algorithm are described in Section II. Section III deals with the problem of optimizing the consensus scheme convergence speed. Characteristic numerical simulations are given Section IV, while concluding remarks are stated in Section V.

II. ADAPTIVE CONSENSUS ALGORITHM

A. Problem Definition

Assume that we have a network of N intelligent sensors, represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where

N. Ilić, who is the corresponding author, is with the College of Applied Technical Sciences, 37000 Kruševac, Serbia and Vlatacom Institute, 11000 Belgrade, Serbia; E-mail: nemili@etf.rs

M. S. Stanković is with the Innovation Center, Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; Vlatacom Institute, 11000 Belgrade, Serbia; COPELABS, Universidade Lusófona, Lisboa, Portugal and Singidunum University, 11000 Belgrade, Serbia; E-mail: milsta@kth.se

S. S. Stanković is with the Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Serbia; E-mail: stankovic@etf.rs

$\mathcal{N} = \{1, \dots, N\}$ is the set of nodes and $\mathcal{E} = \{(i,j)\}$ is the set of directed links from node i to node j. Let A be the adjacency matrix of $\mathcal{G}$. We shall assume that the network is aimed at performing a global information processing task in a distributed manner, based on the locally available information and real-time communication via the available communication links (local connectivity assumption). In order to accomplish the adopted global task, we assume that each node locally processes its state vector:

$$\xi_i(t|t) = \xi_i(t|t-1) + F_i\left(z_i(t)\right), \tag{1}$$

where ξ_i represents the state vector (size $n \times 1$) of the node i, F_i its local filtering function, and z_i locally available measurements. This setting encompasses typical distributed estimation and tracking problems, as well as distributed change detection, optimization and calibration tasks. Importantly, we shall assume that different nodes in the network can be in different real-time situations regarding the availability of measurements (local observability assumption), so that the quality of the individual filtering processes can differ significantly in time. In this respect, we shall further assume that each node can estimate the quality of its local filtering process, resulting in the scalar local weight $\gamma_i(t) \in (0,1]$, reflecting quality of the estimate $\xi_i(t|t)$. These weights $\gamma_i(t)$ typically directly reflect the availability of measurements (as in target tracking scenarios where only part of the nodes observe the target).

The nodes subsequently exchange information on their states and perform local predictions:

$$\xi_i(t+1|t) = P_i\left(\sum_{j\in\mathcal{J}_i} c_{ij}(t)\xi_j(t|t)\right), \tag{2}$$

where P_i is the local prediction function, and $c_{ij}(t)$, $i,j = 1,\dots,N$, are time varying weights reflecting the real-time communication weights between the nodes, such that $N \times N$ matrix $C(t) = [c_{ij}(t)]$ (consensus matrix) is row-stochastic for all t, $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$, where $\mathcal{N}_i$ is the set of in-neighbors of the i-th node, *e.g.*, [1], [26], [27].

We see that the proposed setting requires the exchange of node states only (size $n \times 1$) between the nodes (compare with [22], [19]). It contains two parts: 1) the filtering part, in which the local measurements are processed, and 2) the prediction part, in which the agreement between the nodes is enforced by forming a convex combination of the communicated local states, which are then included in the prediction step, *e.g.*, [1], [4], [26].

B. Multi-Step Scheme

The main idea behind the adaptive consensus scheme discussed in the paper is to use the locally available weights $\gamma_i(t)$ in the design process of the consensus weights $c_{ij}(t)$, so that the corresponding consensus scheme potentiates the states with high $\gamma_i(t)$. To this end, we shall apply the so-called multi-step consensus, and design such a scheme where the asymptotic consensus weight connected to $\xi_i(t|t)$ will be proportional to $\gamma_i(t)$. Moreover, in order to propose a general solution, we shall assume that each node i is associated with the weight w_i, reflecting *a priori* assertions about the quality of that node, so that the desired asymptotic consensus weights will be

$$c_{ij}^{\infty}(t) = \frac{w_j\gamma_j(t)}{\sum_{j'=1}^{N} w_{j'}\gamma_{j'}(t)}. \tag{3}$$

Consequently, after the communication scheme is applied, all the nodes will have estimations that are mostly influenced by the "high quality" nodes, both in terms of *a priori* as well as real-time quality assertions. Of course, in the case of equal *a priori* weights, we will have $c_{ij}^{\infty}(t) = \gamma_j(t)/\sum_{j'=1}^{N}\gamma_{j'}(t)$.

In order to accomplish the set task, we shall apply K consensus steps at each time instant t. In each consensus step, the nodes will exchange the variables based on $\gamma_i(t)$ as well as the local states. The variables will be used for weighting the communicated states. Before the consensus scheme is applied, we need to obtain matrix B, which represents a consensus matrix based on A used in communicating local weights $\gamma_i(t)$ between the nodes, such that

$$\lim_{K\to\infty} B^K = \mathbf{1}w^T, \tag{4}$$

where $\mathbf{1}$ represents a vector of ones of appropriate size, and w *a priori* weight vector $w = [w_1 \cdots w_N]^T$. Such a matrix satisfies $w^T B = w^T$ [4], [1], [14]. It can be shown that this equation has, in principle, infinitely many solutions, using the procedure analogous to the one from [14]. Solving it requires the knowledge of network topology; it is performed only once for a fixed topology (see the following section).

Let $\gamma_i^{[\kappa]}(t)$ and $\xi_i^{[\kappa]}(t|t)$, $\kappa = 1,\dots,K$, be the weight and the state of the i-th node connected to the κ-th consensus step, respectively. We shall start from

$$\gamma_i^{[1]}(t) = \gamma_i(t), \quad \xi_i^{[1]}(t|t) = \xi_i(t|t).$$

In this first consensus step, the nodes exchange $\gamma_i^{[1]}(t)$ and their states $\xi_i^{[1]}(t|t)$. The weights $\gamma_i^{[1]}(t)$ are being exchanged through B, so that the corresponding consensus matrix that defines the weights for the communicated states is

$$C^{[\kappa]}(t) = \left(B \cdot diag\left(\gamma_1^{[\kappa]}(t),\dots,\gamma_N^{[\kappa]}(t)\right)\right)_{rs}, \tag{5}$$

$\kappa = 1$, where $(\cdot)_{rs}$ denotes an operator making the resulting matrix row-stochastic - it divides elements of each row of the argument matrix by the corresponding row sums.

Now that we have the consensus matrix $C^{[\kappa]}(t) = [c_{ij}^{[\kappa]}(t)]$, $i,j = 1,\dots,N$, the local states are obtained by

$$\xi_i^{*[\kappa]}(t|t) = \sum_{j\in\mathcal{J}_i} c_{ij}^{[\kappa]}(t)\xi_j^{[\kappa]}(t|t), \tag{6}$$

where star denotes the states obtained after the consensus step is applied.

For the next consensus step, our design requires that the weight of each node corresponds to the sum of the weights it received previously, so that $\gamma^{[\kappa+1]}(t) = [\gamma_1^{[\kappa+1]}(t) \quad \cdots \quad \gamma_N^{[\kappa+1]}(t)]^T$ becomes:

$$\gamma^{[\kappa+1]}(t) = B \cdot diag\left(\gamma_1^{[\kappa]}(t),\dots,\gamma_N^{[\kappa]}(t)\right)\cdot\mathbf{1} = B\cdot\gamma^{[\kappa]}(t). \tag{7}$$

Also,

$$\xi_i^{[\kappa+1]}(t|t) = \xi_i^{*[\kappa]}(t|t). \tag{8}$$

At this point one can proceed with (5) and (6) for $\kappa + 1$ and likewise repeat the described procedure for the total of K consensus steps.

It is straightforward to show that the consensus parameters $c_{ij}(t)$ in (2) obtained by applying K consensus steps are:

$$[c_{ij}(t)] = C(t) = C^{[K]}(t) \cdot C^{[K-1]}(t) \cdot \ldots \cdot C^{[1]}(t) = \left[\frac{b_{ij}^K \gamma_j^{[1]}(t)}{\sum_{j=1}^n b_{ij}^K \gamma_j^{[1]}(t)} \right], \quad (9)$$

where $[b_{ij}^K] = B^K$ (B to the power of K). Under the usual assumption that $\mathcal{G}$ is strongly connected [4], [14], we select B satisfying (4), and readily obtain:

$$\lim_{K\to\infty} \xi_i^{*[K]}(t|t) = \sum_{j\in\mathcal{J}_i} c_{ij}^{\infty}(t)\xi_j(t|t), \quad (10)$$

where

$$[c_{ij}^{\infty}(t)] = \begin{bmatrix} \frac{w_1\gamma_1(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} & \cdots & \frac{w_N\gamma_N(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} \\ \vdots & \ddots & \vdots \\ \frac{w_1\gamma_1(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} & \cdots & \frac{w_N\gamma_N(t)}{w_1\gamma_1(t)+\cdots+w_N\gamma_N(t)} \end{bmatrix},$$

which represents the desired result from (3)

III. Optimizing Convergence Speed

In order to have a complete algorithm, we have to define a constant matrix B in (5) and (7) according to (4), for a given network topology and choice of w. Since our concern is to achieve *the fastest convergence* to consensus (maximally reducing the communication efforts), the formal problem to be solved is [25]:

$$\begin{aligned} &\text{minimize } \; r(B - \mathbf{1}w^T) \; \text{ with respect to } B \\ &\text{subject to } w^T B = w^T \text{ and } B\mathbf{1} = \mathbf{1}, \end{aligned} \quad (11)$$

where $r(\cdot)$ is equal to either the spectral radius $\rho(\cdot)$ or the spectral norm $\|\cdot\|_S$ (see [25]). It is possible to reduce the number of degrees of freedom by introducing additional, practically justifiable constraints on A.

Let B have equal non-diagonal non-zero elements in each column (an analogous row-wise assumption can be used as well), *i.e.*, $B =$

$$\begin{bmatrix} \sum_{k,k\neq1}(1-\alpha c_k a_{1k}) & \alpha c_2 a_{12} & \cdots & \alpha c_n a_{1n} \\ \alpha c_1 a_{21} & \sum_{k,k\neq2}(1-\alpha c_k a_{2k}) & \cdots & \alpha c_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha c_1 a_{n1} & \alpha c_2 a_{n2} & \cdots & \sum_{k,k\neq n}(1-\alpha c_k a_{nk}) \end{bmatrix},$$

where α represents a scaling factor, c_j are the elements of the normalized vector of column values

$$c = [c_1 \quad \cdots \quad c_N]^T,$$

so that $\mathbf{1}^T c = 1$, and a_{ij} are the elements of the adjacency matrix A of the digraph $\mathcal{G}$.

Formally, first we have to solve the standard equation

$$w^T B = w^T, \quad (12)$$

for the unknown parameters in B under the given topology constraints. It is straightforward to show that (12) reduces to

$$\tilde{L}_w^T c = 0, \quad (13)$$

where $\tilde{L}_w$ is in the form of a weighted Laplacian matrix of the underlying graph $\mathcal{G}$ [28], *i.e.*, $\tilde{L}_w =$

$$\begin{bmatrix} \sum_{k,k\neq1} w_k a_{1k} & -w_2 a_{12} & \cdots & -w_n a_{1n} \\ -w_1 a_{21} & \sum_{k,k\neq2} w_k a_{2k} & \cdots & -w_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -w_1 a_{n1} & -w_2 a_{n2} & \cdots & \sum_{k,k\neq n} w_k a_{nk} \end{bmatrix}.$$

If $\mathcal{G}$ is strongly connected (which is a standard topology assumption within similar contexts [14]), we have $\mathrm{rank}(\tilde{L}_w) = N-1$ [4], [14]. Thus, (13) represents a system of $N-1$ linearly independent equations with N unknowns, which, combined with $\mathbf{1}^T c = 1$, yields a unique solution for c.

On the other hand, it is easy to see that

$$B = I - \alpha \tilde{L}_c, \quad (14)$$

where $\tilde{L}_c$ is in the form of another weighted Laplacian matrix, *i.e.*, $\tilde{L}_c =$

$$\begin{bmatrix} \sum_{k,k\neq1} c_k a_{1k} & -c_2 a_{12} & \cdots & -c_n a_{1n} \\ -c_1 a_{21} & \sum_{k,k\neq2} c_k a_{2k} & \cdots & -c_n a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ -c_1 a_{n1} & -c_2 a_{n2} & \cdots & \sum_{k,k\neq n} c_k a_{nk} \end{bmatrix}.$$

Now, we can directly generalize the results from [25] to conclude that (4) is achieved for

$$0 < \alpha < \frac{2}{\lambda_1(\tilde{L}_c)}, \quad (15)$$

where $\lambda_i(\cdot)$ denotes the i-th largest eigenvalue of the argument matrix. Further, the optimal choice of α in view of (11) is

$$\alpha = \frac{2}{\lambda_1(\tilde{L}_c) + \lambda_{N-1}(\tilde{L}_c)}. \quad (16)$$

It is to be noted that simple bounds that give choices that do not require exact knowledge of the Laplacian spectrum can be used as well [25].

IV. Simulation Examples

We shall consider a sensor network with $N = 10$ nodes, randomly dispersed within a square area, and connected if the corresponding inter-distance is less than half of the side of the square (the so-called Geometric Random Graph topology). In order to show that our proposed methodology is applicable to the general case of digraphs, approximately 25% of two-way connections is randomly made one-way. One obtained realization of such a network, together with its communication topology, is shown in Fig 1. The corresponding adjacency matrix is

$$A = \begin{bmatrix} 0&0&1&1&0&1&1&0&0&1 \\ 1&0&0&1&0&0&1&0&0&0 \\ 1&1&0&0&0&0&1&0&0&0 \\ 1&1&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&0&1&0&0&0&1&1&0&1 \\ 1&1&0&0&0&1&0&0&0&1 \\ 0&0&0&0&1&1&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&1&1&1&0&1&0 \end{bmatrix}.$$

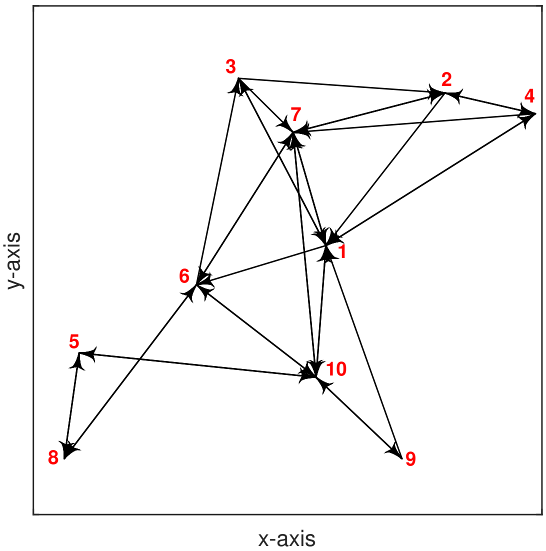

Fig. 1. Sensor network with its communication topology.

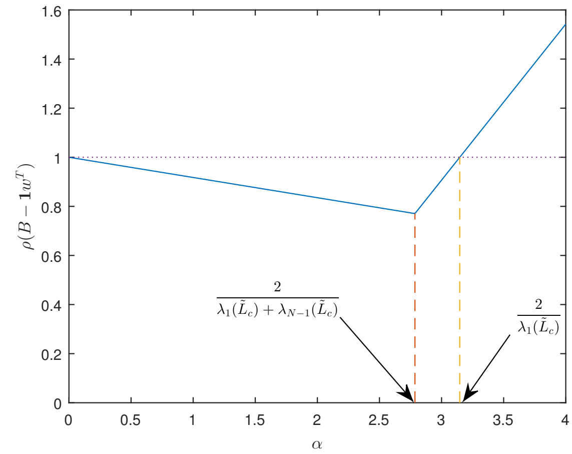

Fig. 2. Spectral radius of $B - \mathbf{1}w^T$ for different values of α.

First, we shall consider the average consensus problem, *i.e.*, the case when we have

$$w = [0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1\ 0.1]^T.$$

The corresponding weighted Laplacian matrix is

$$\tilde{L}_w = \begin{bmatrix} 0.6 & 0 & -0.1 & -0.1 & 0 & -0.1 & -0.1 & 0 & 0 & -0.1 \\ -0.1 & 0.3 & 0 & -0.1 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0.2 & 0 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & -0.1 \\ 0 & 0 & -0.1 & 0 & 0 & 0.4 & -0.1 & -0.1 & 0 & -0.1 \\ -0.1 & -0.1 & 0 & 0 & 0 & -0.1 & 0.6 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & -0.1 & -0.1 & 0 & 0.2 & 0 & 0 \\ -0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.1 \\ -0.1 & 0 & 0 & 0 & -0.1 & -0.1 & -0.1 & 0 & -0.1 & 0.5 \end{bmatrix}.$$

By coupling (13) (for example, taking the first $N-1$ independent rows) with $\mathbf{1}^T c = 1$, we come to

$$\begin{bmatrix} 0.6 & -0.1 & -0.1 & -0.1 & 0 & 0 & -0.1 & 0 & -0.1 & -0.1 \\ 0 & 0.3 & -0.1 & -0.1 & 0 & 0 & -0.1 & 0 & 0 & 0 \\ -0.1 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & 0 & 0 & 0 \\ -0.1 & -0.1 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & -0.1 & 0 & -0.1 \\ -0.1 & 0 & 0 & 0 & 0 & 0.4 & -0.1 & -0.1 & 0 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & 0 & -0.1 & 0.6 & 0 & 0 & -0.1 \\ 0 & 0 & 0 & 0 & -0.1 & -0.1 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

and readily obtain

$$c = [0.08\ 0.08\ 0.11\ 0.11\ 0.10\ 0.09\ 0.06\ 0.09\ 0.18\ 0.10]^T.$$

The corresponding weighted Laplacian matrix is given by $\tilde{L}_c =$

$$\begin{bmatrix} 0.47 & 0 & -0.11 & -0.11 & 0 & -0.09 & -0.06 & 0 & 0 & -0.10 \\ -0.08 & 0.25 & 0 & -0.11 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ -0.08 & -0.08 & 0.22 & 0 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ -0.08 & -0.08 & 0 & 0.22 & 0 & 0 & -0.06 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.19 & 0 & 0 & -0.09 & 0 & -0.10 \\ 0 & 0 & -0.11 & 0 & 0 & 0.36 & -0.06 & -0.09 & 0 & -0.10 \\ -0.08 & -0.08 & 0 & 0 & 0 & -0.09 & 0.35 & 0 & 0 & -0.10 \\ 0 & 0 & 0 & 0 & -0.10 & -0.09 & 0 & 0.19 & 0 & 0 \\ -0.08 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.18 & -0.10 \\ -0.08 & 0 & 0 & 0 & -0.10 & -0.09 & -0.06 & 0 & -0.18 & 0.50 \end{bmatrix}.$$

Now that we have $\tilde{L}_c$, we can design the matrix B from (14) by choosing the scaling factor α. Plotting the spectral radius of $B - \mathbf{1}w^T$ for different values of α (Fig. 2), we see that the minimal value and fastest convergence is obtained exactly for $\alpha = 2/(\lambda_1(\tilde{L}_c) + \lambda_{N-1}(\tilde{L}_c)) = 2.79$, while the stability boundary is reached for $\alpha = 2/\lambda_1(\tilde{L}_c) = 3.15$. The resulting matrix B providing the fastest consensus is

$$B = \begin{bmatrix} -0.31 & 0 & 0.30 & 0.30 & 0 & 0.25 & 0.16 & 0 & 0 & 0.28 \\ 0.22 & 0.31 & 0 & 0.30 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0.22 & 0.23 & 0.39 & 0 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0.22 & 0.23 & 0 & 0.39 & 0 & 0 & 0.16 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.46 & 0 & 0 & 0.26 & 0 & 0.28 \\ 0 & 0 & 0.30 & 0 & 0 & -0.01 & 0.16 & 0.26 & 0 & 0.28 \\ 0.22 & 0.23 & 0 & 0 & 0 & 0.25 & 0.02 & 0 & 0 & 0.28 \\ 0 & 0 & 0 & 0 & 0.27 & 0.25 & 0 & 0.48 & 0 & 0 \\ 0.22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.50 & 0.28 \\ 0.22 & 0 & 0 & 0 & 0.27 & 0.25 & 0.16 & 0 & 0.50 & -0.40 \end{bmatrix}.$$

We shall further consider the weighted consensus case. For example, let the weight vector be

$$w = [0.05\ 0.16\ 0.08\ 0.12\ 0.22\ 0.06\ 0.09\ 0.06\ 0.06\ 0.11]^T.$$

Similarly as above, we obtain

$$c = [0.04\ 0.11\ 0.16\ 0.12\ 0.21\ 0.07\ 0.05\ 0.07\ 0.08\ 0.10]^T,$$

and the fastest convergence is achieved for $\alpha = 3.1$, while the stability boundary is reached for $\alpha = 3.35$. The resulting matrix B providing the fastest consensus is

$$B = \begin{bmatrix} -0.52 & 0 & 0.49 & 0.36 & 0 & 0.21 & 0.15 & 0 & 0 & 0.31 \\ 0.11 & 0.38 & 0 & 0.36 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0.11 & 0.33 & 0.40 & 0 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0.11 & 0.33 & 0 & 0.40 & 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.48 & 0 & 0 & 0.21 & 0 & 0.31 \\ 0 & 0 & 0.49 & 0 & 0 & -0.16 & 0.15 & 0.21 & 0 & 0.31 \\ 0.11 & 0.33 & 0 & 0 & 0 & 0.21 & 0.04 & 0 & 0 & 0.31 \\ 0 & 0 & 0 & 0 & 0.65 & 0.21 & 0 & 0.14 & 0 & 0 \\ 0.11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.57 & 0.31 \\ 0.11 & 0 & 0 & 0 & 0.65 & 0.21 & 0.15 & 0 & 0.25 & -0.38 \end{bmatrix}.$$

In order to show the effectiveness of our solution, we shall compare the aggregated consensus results (powers of the matrix B) in the optimized convergence speed case (with $\alpha = 2/(\lambda_1(\tilde{L}_c)+\lambda_{N-1}(\tilde{L}_c))$, shown in Fig. 3), to one example of the non-optimized case (with $\alpha = 1$, shown in Fig. 4). It can be clearly seen that the desired asymptotic weights are reached much faster with the optimized solution.

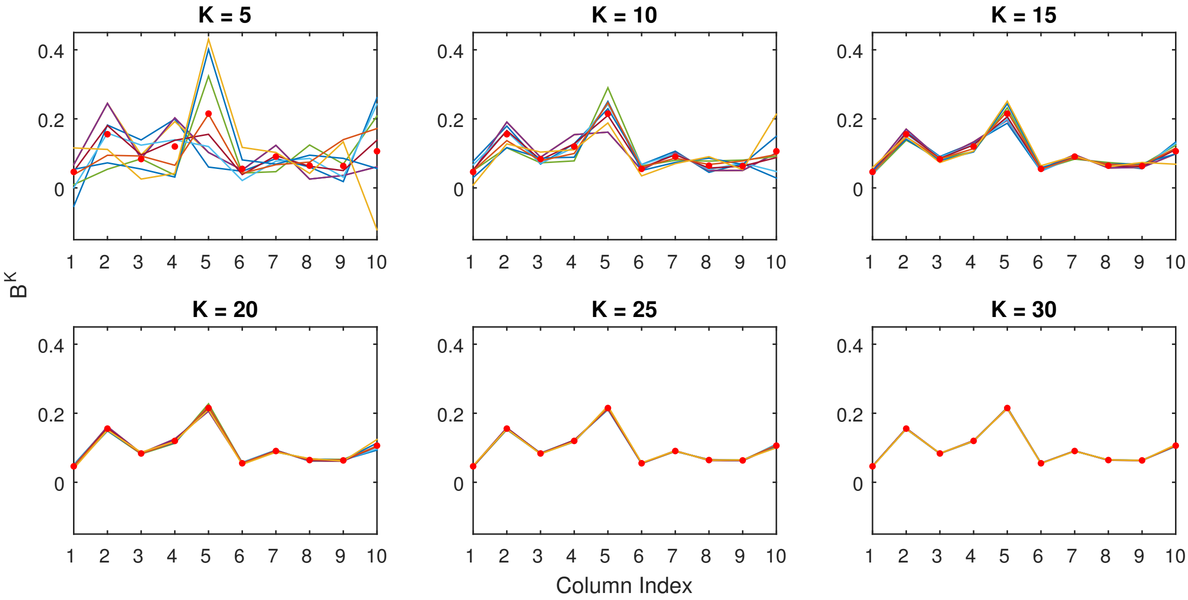

Fig. 3. Values of all rows of B^K when B is optimized; asymptotic weights w are illustrated with red dots.

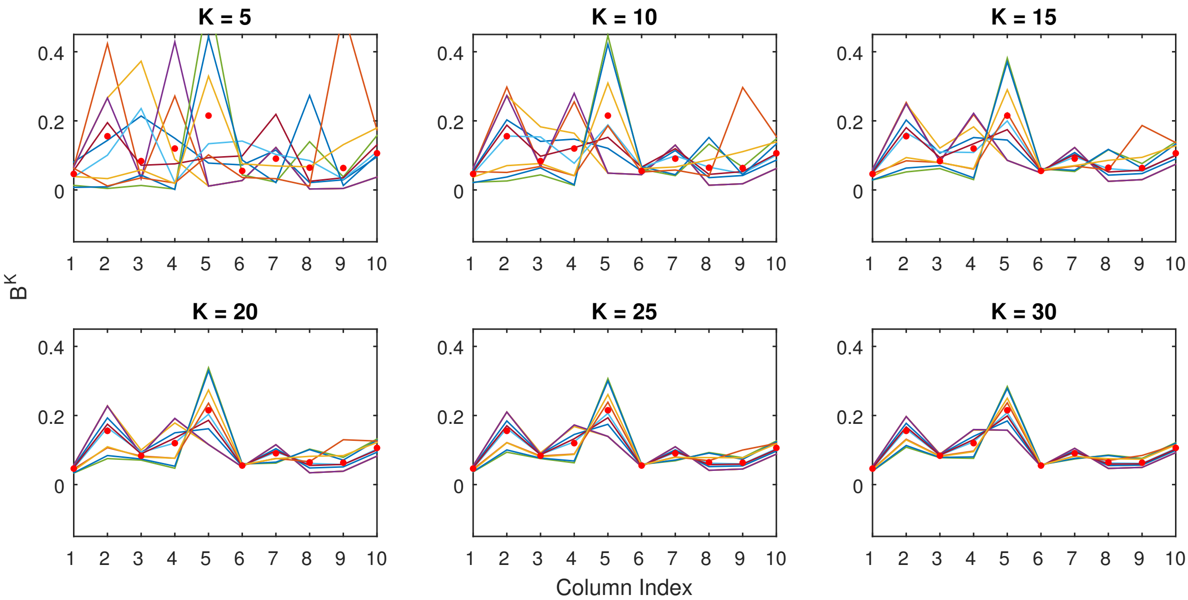

Fig. 4. Values of all rows of B^K when B is not optimized; asymptotic weights w are illustrated with red dots.

Finally, as a small proof-of-concept example, we shall assume that the nodes states directly reflect the received scalar measurements; for all the nodes except nodes 5 and 8 these are randomly sampled from the Gaussian distribution with $\mu = 100$ and $\sigma = 10$ (we assume that these are low quality measurements with corresponding weights $\gamma_i = 0.001$), while for nodes 5 and 8 the measurements are randomly sampled from the Gaussian distribution with $\mu = 10$ and $\sigma = 1$ (high quality measurements with corresponding weights $\gamma_i = 0.99$). We adopt equal *a priori* weights w_i for all i. Therefore, we expect our scheme to achieve consensus value around 10, influenced primarily by nodes 5 and 8. We see in Figs. (5) and (6) that this indeed happens. Also, it can be seen that the optimized case offers a satisfying solution already for a number of consensus steps of 3, while similar amount of disagreement between the nodes in the non-optimized case is reached only just for a total number of consensus steps of 10.

V. Conclusion

In this paper a procedure for designing the adaptive multi-step consensus communication scheme in sensor networks is

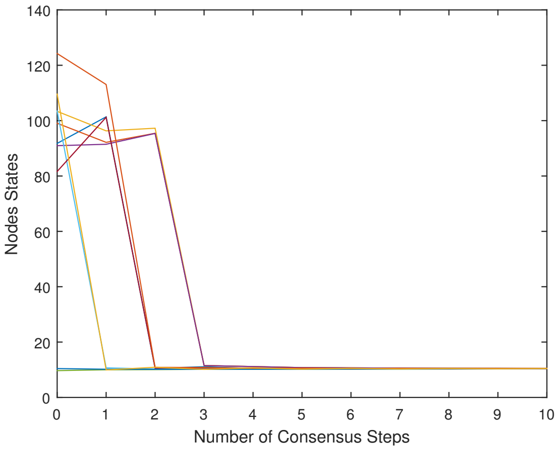

Fig. 5. Nodes' states for different number of consensus steps - optimized case.

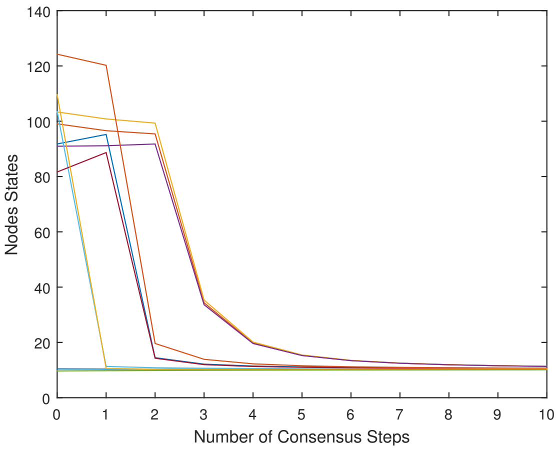

Fig. 6. Nodes' states for different number of consensus steps - non-optimized case.

proposed, exhibiting the fastest convergence to the desirable asymptotic values, and thus minimizing the resulting communication burden. The scheme allows both *a priori* and real-time weightings of the local processing results connected to different nodes in the network, influencing the asymptotic behavior of the consensus protocol in appropriate manner.

Further work can be oriented towards exploring the design possibilities of analogous randomized schemes, similarly as in [29], [12], [14].

References

[1] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[2] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[3] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*. Springer Publishing Company, Incorporated, 2007.

[4] W. Ren and R. Beard, "Consensus seeking in multi-agent systems using dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, 2005.

[5] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," in *American Control Conference*, 2008, pp. 3157–3162.

[6] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimator," *IEEE Trans. Autom. Control*, vol. 54, pp. 410–415, 2009.

[7] ——, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

[8] R. M. G. Ferrari, T. Parisini, and M. L. Polycarpou, "Distributed fault diagnosis with overlapping decompositions and consensus filters," in *Proc. Amer. Contr. Conf.*, 2007.

[9] N. Ilić, M. Stanković, and S. Stanković, "Consensus based overlapping decentralized observer for fault detection and isolation," in *Proc. Melecon 2010 Conf.*, 2010.

[10] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 56, pp. 3375–3380, 2008.

[11] P. Braca, S. Marano, V. Matta, and P. Willett, "Asymptotic optimality of running consensus in testing binary hypotheses," *IEEE Trans. Signal Processing*, vol. 58, pp. 814–825, 2010.

[12] S. S. Stanković, N. Ilić, M. S. Stanković, and K. H. Johansson, "Distributed change detection based on a consensus algorithm," *IEEE Transactions on Signal Processing*, vol. 59, pp. 5686–5697, 12 2011.

[13] N. Ilić, S. S. Stanković, M. Stanković, and K. H. Johansson, "Consensus based distributed change detection using generalized likelihood ratio methodology," *Signal Processing*, vol. 92, 06 2011.

[14] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4069–4074, 2016.

[15] N. Ilić, M. S. Stanković, and S. S. Stanković, "Adaptive consensus-based distributed target tracking in sensor networks with limited sensing range," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 778–785, 2014.

[16] K. O. A. Ali, N. Ilić, M. S. Stanković, and S. S. Stanković, "Consensus-based distributed adaptive target tracking in camera networks using integrated probabilistic data association," *EURASIP Journal on Advances in Signal Processing*, vol. 2018, no. 1, p. 13, 2018. [Online]. Available: https://doi.org/10.1186/s13634-018-0534-z

[17] ——, "Distributed target tracking in sensor networks using multi-step consensus," *IET Radar, Sonar Navigation*, vol. 12, no. 9, pp. 998–1004, 2018.

[18] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[19] A. T. Kamal, J. H. Bappy, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed multi target tracking and data association in vision networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1397–1410, 2016.

[20] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013.

[21] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, Sept 2004.

[22] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *47th IEEE Conference on Decision and Control*, 2008, pp. 1085–1090.

[23] N. Ilić, K. O. Al-Ali, M. Stanković, and S. Stanković, "Distributed multi-target tracking in camera networks using multi-step consensus," in *Proc. 4th IcETRAN Conf.*, 2017.

[24] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Review*, vol. 53, no. 4, pp. 747–772, 2011. [Online]. Available: https://doi.org/10.1137/110837462

[25] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[26] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, pp. 803–812, 1986.

[27] R. A. Horn and C. A. Johnson, *Matrix Analysis*. Cambridge, England: Cambridge University Press, 1985.

[28] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer Verlag, 2001.

[29] N. Ilić and S. S. Stanković, "Communication gains design in a consensus based distributed change detection algorithm," in *Proc. 8th European Workshop on Advanced Control and Diagnosis*, 2010.