HIGHER DIMENSIONAL CONSENSUS ALGORITHMS IN SENSOR NETWORKS

Usman A. Khan, Soummya Kar and José M. F. Moura

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh, PA
{ukhan, moura}@ece.cmu.edu, {soummyak@andrew.cmu.edu}

***Abstract*— This paper introduces higher dimensional consensus, a framework to capture a number of different, but, related distributed, iterative, linear algorithms of interest in sensor networks. We show that, by suitably choosing the iteration matrix of the higher dimensional consensus, we can capture, besides the standard average-consensus, a broad range of applications, including sensor localization, leader-follower, and distributed Jacobi algorithm. We work with the concept of anchors and explicitly derive the consensus subspace and provide the dimension of the limiting state of the sensors.**

***Index Terms*—** Distributed algorithms, Iterative methods, Distributed control, Networks, Large-scale systems

I. Introduction

There is a renewed interest in the sensor network community on consensus algorithms. In consensus, the sensors reach a common state, in a distributed and iterative fashion with only local communication and computation. Consensus is iterative because the information fuses over a sparsely connected sensor network. Applications for sensor networks include, but are not restricted to load balancing [1], multi-vehicle control and navigation [2], flocking [3], sensor localization [4].

Early work includes [5], [6]. Much of the existing work is focused on average-consensus problems where the goal is to drive the state of the sensors to the average of their initial states [7], [8]. Its generalizations and extensions to random environments, quantized information exchange, gossip protocols, convergence issues and topology optimization can also be found in the literature [9], see [10] for a detailed set of references.

In this paper, we present a general consensus algorithm that captures average consensus as a special case. We term this higher dimensional consensus (HDC) algorithms. Our framework accounts for two types of nodes: anchors [11] and sensors (in strict sense). Anchors are sensors that do not update their state and behave as leaders, providing an appropriate frame of reference specific to the pertinent application. Anchors, thus, provide additional degrees of freedom to the underlying algorithm and form a building block to generalize consensus algorithms to include in the same set-up, besides traditional average consensus, applications like distributed sensor localization [4], solving linear system of equations using Jacobi algorithms [10], and leader-follower architectures [10].

Key to our approach is the notion of dimension of the consensus subspace. We show that the limit state of the sensors resides in a subspace that can be at most n-dimensional, where n is the number of the anchors. HDC algorithms also play an important role in the assimilation of local covariances for distributed estimators. The estimators are implemented on low-order local subsystems obtained by spatially decomposing a large-scale dynamical system [12].

We summarize the rest of the paper. Section II contains background on distributed algorithms, whereas Section III gives our formulation of the problem. Section IV discusses the consensus algorithm in higher dimensions, and Section V illustrates our approach with specific applications in the leader-follower scenario. We discuss robustness issues in the presence of chaotic environment in Sections VI and conclude the paper in Section VII.

II. Background

Consider a network with N sensors. The state of an arbitrary sensor, l, is denoted by a scalar[1], c_l. A typical linear, iterative, distributed algorithm is

$$c_l(t+1) = \sum_{j \in \mathcal{K}(l)} v_{lj} c_j(t), \qquad 1 \leq l \leq N, \tag{1}$$

where $\mathcal{K}(l)$ denotes the neighboring sensors of sensor l, i.e., those sensors that can send information to sensor l, and v_{lj} are the state-updating coefficients in the linear combinations. The goal of the linear algorithm (1) is for the state of the sensor network to converge to

$$\lim_{t \to \infty} c_l(t+1) = f\left(c_1(0), c_2(0), \ldots, c_N(0)\right), \tag{2}$$

where $f(\cdot)$ denotes a particular linear function of the initial states. For example, $f(\cdot)$ can be the average of the initial states in the average-consensus problem. In this paper, we look for more flexibility. This is achieved by introducing anchors, sensors whose states do not change.

We now write (1) in matrix format. Let $\mathbf{c}(t)$ denote the $N \times 1$ vector that collects the state of the network at time t, i.e.,

$$\mathbf{c}(t) = \left[c_1(t), \ldots, c_N(t)\right]^T. \tag{3}$$

The matrix form of (1) is

$$\begin{aligned} \mathbf{c}(t+1) &= \boldsymbol{\Upsilon}\mathbf{c}(t), \\ &= \boldsymbol{\Upsilon}^{t+1}\mathbf{c}(0), \end{aligned} \tag{4}$$

[1]The analysis can be easily extended to m-dimensional states, for details, see [4].

where $\boldsymbol{\Upsilon} = \{\upsilon_{lj}\}$ is the $N \times N$ iteration matrix that collects the coefficients, υ_{lj}. The iteration matrix, $\boldsymbol{\Upsilon}$, is sparse and its sparsity (non-zero) pattern reflects the underlying communication graph. The iteration matrix, $\boldsymbol{\Upsilon} = \{\upsilon_{lj}\}$, can be tuned to the particular application of interest under the constraints that (i) the sparsity pattern of the iteration matrix, $\boldsymbol{\Upsilon}$, does not violate the underlying communication graph; and (ii) the limit,

$$\lim_{t\to\infty} \boldsymbol{\Upsilon}^{t+1}, \tag{5}$$

exists. We will see that under these conditions, we can choose the elements, υ_{lj}, of the iteration matrix, $\boldsymbol{\Upsilon}$, to achieve, in a distributed fashion, a desired task.

III. Problem Formulation

We partition the N nodes in the network into M sensors and n anchors, i.e., $N = M + n$. The set of anchors, κ, contains the nodes that do not change their states. The anchors are the drivers or leaders of the system that take the algorithm in the direction of accomplishing a particular task. The set of sensors, Ω, contains the sensors whose states change after each iteration[2]. We index the set of anchors, κ, by $1, \ldots, n$ and the set of sensors, Ω, by $n+1, \ldots, N$. Let $u_k(t)$ denote the state of the kth anchor at time t. Since anchors do not update their state, we have

$$u_k(t+1) = u_k(t) = u_k(0), \qquad t \geq 0,\ k \in \kappa. \tag{6}$$

Similarly, let $x_l(t)$ denote the state of the lth sensor at time t. Define

$$\mathbf{u}(0) = [u_1(0), \ldots, u_n(0)]^T, \tag{7}$$

$$\mathbf{x}(t) = [x_{n+1}(t), \ldots, x_N(t)]^T. \tag{8}$$

We partition the state of the entire sensor network, $\mathbf{c}(t)$, as

$$\mathbf{c}(t) = \begin{bmatrix} \mathbf{u}(0) \\ \mathbf{x}(t) \end{bmatrix}. \tag{9}$$

Similarly, we partition the $N \times N$ iteration matrix, $\boldsymbol{\Upsilon}$, as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{B} & \mathbf{P} \end{bmatrix}. \tag{10}$$

The general form of the higher dimension consensus (HDC) algorithm, thus, becomes

$$\mathbf{x}(t+1) = \mathbf{P}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(0), \tag{11}$$

where: $\mathbf{I}_n$ is the $n \times n$ identity matrix; the $M \times n$ matrix, $\mathbf{B}$, collects the updating coefficients for the sensors in terms of the anchors; and the $M \times M$ matrix, $\mathbf{P}$, collects the updating coefficients for the sensors in terms of the sensors themselves.

We write (4) in terms of the network initial conditions,

$$\mathbf{x}(t+1) = \mathbf{P}^{t+1}\mathbf{x}(0) + \sum_{k=0}^{t} \mathbf{P}^k \mathbf{B}\mathbf{u}(0). \tag{12}$$

Given this setup, we now analyze the convergence conditions of (11) and discuss relevant practical applications where the algorithm (11) is useful.

[2]In the sequel, a sensor implies a non-anchor node.

IV. Consensus in Higher Dimensions

Before introducing consensus in higher dimensions, we briefly sketch the traditional average-consensus problem. To proceed with the discussion, let $\rho(\mathbf{P})$ be the spectral radius of $\mathbf{P}$, defined as

$$\rho(\mathbf{P}) = \max_i |\lambda_i(\mathbf{P})|, \tag{13}$$

where $\lambda_i(\mathbf{P})$ is the ith eigenvalue of $\mathbf{P}$.

Average-consensus: In average-consensus, $\mathbf{B} = \mathbf{0}$ in (11) and the algorithm reduces to

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{P}\mathbf{x}(t), \\ &= \mathbf{P}^{t+1}\mathbf{x}(0). \end{aligned} \tag{14}$$

Consensus converges to the average of sensors' initial conditions if

$$\rho(\mathbf{P}) = 1, \tag{15}$$

and, we get

$$\lim_{t\to\infty} \mathbf{P}^{t+1} = \frac{\mathbf{1}\mathbf{1}^T}{M}, \tag{16}$$

under some additional conditions on $\mathbf{P}$. In the above equation, $\mathbf{1}$ is the $M \times 1$ column vector of 1's and M is the number of sensors. These and other conditions under which (16) holds are very well-studied, see, for instance, [2].

Higher dimensional consensus ($n \geq 1$): Now, we extend the average-consensus problem to consensus in higher dimensions. We consider $\mathbf{B} \neq \mathbf{0}$ that also implies at least one anchor. We establish the convergence of the HDC algorithm (11) in the following lemma.

Lemma 1: Let $\mathbf{x}_\infty$ denote the limit state of the sensors, i.e.,

$$\mathbf{x}_\infty = \lim_{t\to\infty} \mathbf{x}(t+1). \tag{17}$$

If

$$\rho(\mathbf{P}) < 1, \tag{18}$$

then

$$\mathbf{x}_\infty = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}\mathbf{U}(0). \tag{19}$$

Proof: The proof is a consequence of (12) and Lemma 3 in Appendix I. ∎

The limit state of the sensors, $\mathbf{x}_\infty$, is independent of the sensors' initial conditions, i.e., the algorithm forgets the sensors' initial conditions. In other words, it converges to (19) for arbitrary sensors' initial conditions. Let

$$\mathbf{W} = (\mathbf{I} - \mathbf{P})^{-1}\mathbf{B}. \tag{20}$$

The following lemma establishes rank conditions on (20).

Lemma 2: Let $n < M$ and $r_{\mathbf{W}}$ denote the rank of $\mathbf{W}$. A necessary condition for (20) to hold is

$$r_{\mathbf{B}} = r_{\mathbf{W}}. \tag{21}$$

Proof: Let $\mathbf{Q} = (\mathbf{I} - \mathbf{P})^{-1}$, then $r_{\mathbf{Q}} = M$. By hypothesis $n < M$, and from Lemma 4 in Appendix I, we have

$$\text{rank}(\mathbf{QB}) \leq r_{\mathbf{B}}, \tag{22}$$

$$\text{rank}(\mathbf{QB}) \geq M + r_{\mathbf{B}} - M = r_{\mathbf{B}}. \tag{23}$$

The condition (21) now follows, since from (20), we also have

$$\text{rank}(\mathbf{QB}) = r_{\mathbf{W}}. \tag{24}$$

■

A. Consensus subspace

We define the consensus subspace as follows.

Definition 1 (Consensus subspace): Given the matrices, $\mathbf{B} \in \mathbb{R}^{M\times n}$ and $\mathbf{P} \in \mathbb{R}^{M\times M}$, such that $\rho(\mathbf{P}) < 1$, the consensus subspace, Ξ, is defined as

$$\Xi = \{\mathbf{x}_\infty \mid \mathbf{x}_\infty = (\mathbf{I}-\mathbf{P})^{-1}\mathbf{B}\mathbf{u}(0),\ \mathbf{u}(0) \in \mathbb{R}^n\}. \tag{25}$$

The dimension of Ξ is established in the following theorem.

Theorem 1: If $1 \le n < M$ and $\rho(\mathbf{P}) < 1$, then the dimension of the consensus subspace, Ξ, is

$$\dim(\Xi) = \text{rank}(\mathbf{B}) \le n. \tag{26}$$

Proof: The proof follows From Lemma 1 and Lemma 2. ■

Now, we formally define the dimension of the higher dimension consensus (HDC) algorithm (11).

Definition 2 (Dimension): The dimension of the HDC algorithm is the dimensions of the consensus subspace, Ξ, given by

$$\dim(\Xi) = \text{rank}(\mathbf{B}). \tag{27}$$

Remarks: Here, we make some additional comments. The sensor localization problem in m-dimensional Euclidean space can be cast as HDC algorithm. In this setting, the state of each sensor is its m-dimensional position estimate that is an $m \times 1$ row vector. The entire exposition in this paper follows for the case of row-vector states. The HDC algorithm with appropriate choices of $\mathbf{B}$ and $\mathbf{P}$ can be formulated such that the limit state is the exact sensor positions [4].

In the next section, we illustrate how to design the matrices $\mathbf{B}$ and $\mathbf{P}$ to fit the leader-follower application. Other designs lead to significant practical applications, for instance, distributed sensor localization [4], Jacobi algorithm [10], among others.

V. Leader-Follower Algorithm

To capture the leader-follower algorithm, we assume that the network has $n \ge 1$ anchors. The goal is that the sensor states converge to the state of the anchor (if $n = 1$) or converge to a linear combination of the states of the anchors (when $n > 1$). We characterize leader-follower algorithms into the following two categories.

A. One anchor, $n = 1$

In this case, we have only one anchor, $n = 1$, whose fixed state is denoted by u_1. The goal is for the entire sensor network to converge to the state, u_1, of the anchor. Let $\mathbf{1}_M$ be the $M\times 1$ column vector of 1's; then we would like the state of the entire sensor network, asymptotically, to be

$$\lim_{t\to\infty} \mathbf{x}(t+1) = \mathbf{1}_M u_1. \tag{28}$$

Since $n = 1$, the iteration matrix, $\boldsymbol{\Upsilon}$, can be partitioned as

$$\boldsymbol{\Upsilon} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{b} & \mathbf{P} \end{bmatrix}, \tag{29}$$

where $\mathbf{b}$ is an $M \times 1$ column vector.

1) Design of the iteration matrix, $\boldsymbol{\Upsilon}$: Relating (28) to (19), we require

$$(\mathbf{I}-\mathbf{P})^{-1}\mathbf{b} = \mathbf{1}_M. \tag{30}$$

We can write (30) as

$$\begin{aligned} \mathbf{b} &= (\mathbf{I}-\mathbf{P})\,\mathbf{1}_M, \\ &= \mathbf{1}_M - \mathbf{P}\mathbf{1}_M. \end{aligned} \tag{31}$$

Note that, due to (18), the matrix $\mathbf{I}-\mathbf{P}$ is invertible, i.e., its eigenvalues, $1-\lambda_i(\mathbf{P})$, can not be 0. Let $\mathbf{b} = [b_{11},\ \ldots,\ b_{M1}]^T$. Element-wise, we have the following condition on $\mathbf{b}$ and $\mathbf{P}$,

$$b_{l1} \quad + \quad \sum_{j=1}^{M} p_{lj} = 1,\ \ l = 1, \cdots, M, \tag{32}$$

in addition to (18). Note that the above equation simply states that each row of the iteration matrix, $\boldsymbol{\Upsilon}$, sums to 1.

2) Example: We may choose the following design strategy. If sensor l is connected to anchor 1, we choose

$$p_{lj} \quad = \quad b_{l1} = \frac{1}{|\mathcal{K}(l)|}, \qquad j \in \mathcal{K}(l),$$

where $\mathcal{K}(l)$ is the set of neighbors of sensor l and $|\cdot|$ denotes the cardinality of a set. If sensor l is not connected to anchor 1, we choose

$$\begin{aligned} p_{lj} &= \frac{1}{|\mathcal{K}(l)|}, \qquad j \in \mathcal{K}(l), & (33)\\ b_{l1} &= 0. & (34) \end{aligned}$$

It can be noted that, under the assumption of strong connectivity of the sensor communication graph and with the above choice, the matrix, $\mathbf{P}$, is irreducible and further strictly substochastic guaranteeing (18), for details, see [10].

B. Multiple anchors, $n > 1$

When we have multiple anchors ($n > 1$), convergence to an arbitrary linear combination of the anchors' initial condition may not be possible since the matrices, $\mathbf{B}$ and $\mathbf{P}$, have sparsity constraints, i.e., they have to follow the adjacency matrix of the underlying sensor communication graph. From (20), it follows that

$$\mathbf{B} = \mathbf{W} - \mathbf{P}\mathbf{W}, \tag{35}$$

which along with (18) and (21) becomes the design criteria.

Example: Let

$$w_i \in [0,1], \qquad \sum_i w_i = 1,\ i = 1, \ldots, n. \tag{36}$$

For all of the sensors to converge to the same convex combination of the anchors, we take the weight matrix, $\mathbf{W}$, to be of the form

$$\mathbf{W} = \begin{bmatrix} w_1 & w_2 & \dots & w_n \\ & \vdots & & \\ w_1 & w_2 & \dots & w_n \end{bmatrix}. \quad (37)$$

It follows that rank $(\mathbf{B}) = 1$, since rank $(\mathbf{W}) = 1$, i.e., all columns of $\mathbf{B}$ should be linearly dependent. Mathematically[3],

$$b_{lk} \neq 0, \text{ for any } k \in \kappa \Rightarrow b_{lk} \neq 0, \forall\ k \in \kappa, \quad (38)$$

$$b_{lk} = 0, \text{ for any } k \in \kappa \Rightarrow b_{lk} = 0, \forall\ k \in \kappa. \quad (39)$$

Equation (38) clearly says that, if any sensor, l, communicates to an anchor, k, then it has to communicate to all the anchors, which may not be possible for a given sensor communication topology. Furthermore, let $\mathbf{b}_k$ denote the kth column of the matrix $\mathbf{B}$; then for every $k, j \in \kappa$ there exists some $\beta_{kj} \in \mathbb{R}$, such that

$$\mathbf{b}_k = \beta_{kj}\mathbf{b}_j. \quad (40)$$

More interesting cases arise when rank$(\mathbf{W})$ is full, for example, in sensor localization and solving linear systems of equations in sensor networks, see [4],[10].

VI. Robustness

Robustness is key in the context of these algorithms when the information exchange is subject to communication noise, packet drops, and imprecise knowledge of system parameters. We propose a modification to the HDC algorithms along the lines of the Robbins-Monro algorithm where the iterations are performed with a decreasing step-size sequence that satisfies some persistence conditions. With such step-sizes, we have shown almost sure convergence of the sensor localization algorithm to their exact locations under broad random phenomenon [4]. This modification is easily extended to the general class of HDC algorithms.

VII. Conclusions

In this paper, we presented a unifying view of commonly used linear distributed iterative algorithms in sensor networks. The notion of "consensus" has been described in a broader framework in higher dimensions that allows us to use the same set of tools for a broad variety of problems: average-consensus, multi-agent coordination, sensor localization, and distributed algorithms to solve linear systems of algebraic equations. Our generalized framework also provides a basis to analyze such algorithms for convergence and robustness. We illustrated our approach with the leader-follower application.

[3]We assume that $\mathbf{b}_k \neq \mathbf{0}, \forall\ k$. If this is not true, then the kth anchor does not send information to any sensor and, hence, does not take part in the algorithm. So it can be removed from the analysis.

Appendix I
Important Results

Lemma 3: If a matrix $\mathbf{P}$ is such that

$$\rho(\mathbf{P}) < 1,$$

where $\rho(\cdot)$ denotes the spectral norm of a matrix, then

$$\lim_{t\to\infty} \mathbf{P}^{t+1} = \mathbf{0}, \quad (41)$$

$$\lim_{t\to\infty} \sum_{k=0}^{t} \mathbf{P}^k = (\mathbf{I} - \mathbf{P})^{-1}. \quad (42)$$

Proof: The proof is straightforward. ■

Lemma 4: Let r_A be the rank of the $M \times M$ matrix $(\mathbf{I} - \mathbf{P})^{-1}$, and r_B the rank of the $M \times n$ matrix $\mathbf{B}$, then

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B} \leq \min(r_A, r_B), \quad (43)$$

$$\text{rank}(\mathbf{I} - \mathbf{P})^{-1}\mathbf{B} \geq r_A + r_B - M. \quad (44)$$

Proof: The proof is available on pages $95-96$ in [13]. ■

Acknowledgements

This work was partially supported by NSF under grants # ECS-0225449 and # CNS-0428404, by ONR under grant # MURI-N000140710747, and by an IBM Faculty Award.

References

[1] G. V. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal on Parallel and Distributed Computing*, vol. 7, pp. 279–301, 1989.

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

[3] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[4] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, Dec. 2008, accepted for publication.

[5] J. N. Tsitsiklis, *Problems in Decentralized Decision Making and Computation*, Ph.D., Massachusetts Institute of Technology, Cambridge, MA, 1984.

[6] J. N. Tsitsiklis and M. Athans, "On the complexity of decentralized decision making and detection problems," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 5, pp. 440–446, May 1985.

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

[8] R. Olfati-Saber and J. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *44th IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005, pp. 6698 – 6703.

[9] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Quantized data," Submitted for publication, 30 pages, see http://arxiv.org/abs/0712.1609, November 2007.

[10] U. Khan, S. Kar, and J. M. F. Moura, "Distributed algorithms in sensor networks," in *Handbook on Sensor and Array Processing*, Simon Haykin and K. J. Ray Liu, Eds. Wiley-Interscience, New York, NY, 2009, to appear, 33 pages.

[11] A. Rahmani and M. Mesbahi, "Pulling the strings on agreement: Anchoring, controllability, and graph automorphism," in *American Control Conference*, New York City, NY, July 11-13 2007, pp. 2738–2743.

[12] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Transactions on Signal Processing*, vol. 56, Part 1, no. 10, pp. 4919–4935, October 2008, DOI: 10.1109/TSP.2008.927480.

[13] G. E. Shilov and R. A. Silverman, *Linear Algebra*, Courier Dover Publications, 1977.