Automatica 50 (2014) 3131–3138

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Robust discrete time dynamic average consensus ☆

Eduardo Montijano [a,1], Juan Ignacio Montijano [b], Carlos Sagüés [c], Sonia Martínez [d]

[a] Centro Universitario de la Defensa (CUD) and Instituto de Investigación en Ingeniería de Aragón (I3A), Zaragoza, Spain
[b] Instituto Universitario de Matemáticas y Aplicaciones (IUMA), Universidad de Zaragoza, Spain
[c] Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
[d] Department of Mechanical and Aerospace Engineering, University of California at San Diego, United States

ARTICLE INFO

Article history:
Received 30 August 2013
Received in revised form
3 March 2014
Accepted 24 June 2014
Available online 4 November 2014

Keywords:
Dynamic average consensus
Multi-agent systems
Robust distributed algorithms

ABSTRACT

This paper deals with the problem of average consensus of a set of time-varying reference signals in a distributed manner. We propose a new class of discrete time algorithms that are able to track the average of the signals with an arbitrarily small steady-state error and with robustness to initialization errors. We provide bounds on the maximum step size allowed to ensure convergence to the consensus with error below the desired one. In addition, for certain classes of reference inputs, the proposed algorithms allow arbitrarily large step size, an important issue in real networks, where there are constraints in the communication rate between the nodes. The robustness to initialization errors is achieved by introducing a time-varying sequence of damping factors that mitigates past errors. Convergence properties are shown by the decomposition of the algorithms into sequences of static consensus processes. Finally, simulation results corroborate the theoretical contributions of the paper.

1. Introduction

This paper studies the problem of reaching the average of a set of time-varying reference signals in a distributed manner, the so called distributed dynamic average consensus. In this problem, each node of the network measures a different time-varying signal and the objective is that agents track the average of all measured signals. Solutions to this problem find numerous applications in diverse fields such as sensor fusion, Spanos, Olfati-Saber, and Murray (2005); cooperative control, Ren (2007); decision making with changing opinions, Montijano, Martínez, and Sagues (2011); and Kalman filtering, Olfati-Saber (2007).

Most of the solutions in the literature consider continuous-time algorithms. Frequency domain analysis is used to guarantee zero steady-state error of ramp inputs in Spanos et al. (2005). The approach presented in Ren (2007) considers a common reference input for all the nodes in the network. A PI-dynamic consensus algorithm is presented in Freeman, Yang, and Lynch (2006) and posteriorly extended in Bai, Freeman, and Lynch (2010) and high order continuous-time dynamic average consensus protocols are investigated in Nosrati, Shafiee, and Menhaj (2009). A good property of these algorithms is their natural robustness against initialization errors. Chen, Cao, and Ren (2012) propose a discontinuous control algorithm able to track bounded signals with bounded derivatives. Non-linear protocols with bounded steady-state error are defined in Nosrati, Shafiee, and Menhaj (2012). Recently, Kia, Cortés, and Martínez (2013) have introduced continuous-time algorithms to solve the dynamic consensus problem. Although all these approaches can be discretized using, e.g., Euler method, the step size they can afford is usually limited.

Discrete-time approaches are more appealing in this regard because they usually can handle larger step-sizes and, thus, have lower communication requirements. To the best of the authors knowledge, the only pure discrete-time approaches (i.e., they do not arise from a discretization) are the ones in Yuan, Liu, Murray, and Gonçalves (2012) and Zhu and Martínez (2010). The convergence analysis of Zhu and Martínez (2010) relies on input-to-output stability, providing bounds on the step size the nodes can choose to guarantee a desired steady-state error with respect to the average. The approach in Yuan et al. (2012) is able to reach dynamic consensus in minimal time, provided that the conditions on the step size given in Zhu and Martínez (2010) are satisfied and the

☆ This work was funded by the Spanish projects Ministerio de Economía, Competitividad y Unión Europea DPI2009-08126 and DPI2012-32100 and project AFOSR-11RSL548. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Xiaobo Tan under the direction of Editor Miroslav Krstic.

E-mail addresses: emonti@unizar.es (E. Montijano), monti@unizar.es (J.I. Montijano), csagues@unizar.es (C. Sagüés), soniamd@ucsd.edu (S. Martínez).
[1] Tel.: +34 976739881; fax: +34 976 739 824.

http://dx.doi.org/10.1016/j.automatica.2014.10.005
0005-1098/© 2014 Elsevier Ltd. All rights reserved.

communication graph is fixed. Unfortunately, these approaches are not robust to initialization errors or measurement noise, which implies they will fail in presence of disturbances.

In this paper we overcome this limitation by proposing a new algorithm inspired by the one in Zhu and Martínez (2010). Our contribution is a new class of discrete time algorithms that are able to reach the dynamic average consensus of a large set of functions with robustness to initialization errors by introducing a time-varying sequence of damping factors that mitigates past errors. In addition, for certain classes of reference inputs, the proposed algorithms allow arbitrarily large step size. For the convergence analysis we decompose our algorithms into a sum of static consensus processes, analyzing the convergence of each of them by means of the eigenvalues of the weight matrices. Finally, we demonstrate the performance of our algorithms with simulations.

The rest of the paper is organized as follows: In Section 2 we introduce the dynamic average consensus problem. Section 3 describes the robust kth order dynamic average consensus (RKODAC) algorithm. We discuss the design of the damping factors of the algorithms in Section 4. Some simulations are shown in Section 5. Finally, the conclusions of this work are in Section 6.

2. Preliminaries and problem statement

We consider a sensor network of N nodes labeled by $i \in \mathcal{V} = \{1, \ldots, N\}$. Communications between the nodes are defined according to a fixed undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents the edge set. In this way, nodes i and j can communicate if and only if $(i, j) \in \mathcal{E}$. Along the paper we assume that the communication graph is connected. The set of neighbors of node $i \in \mathcal{V}$ is the subset of nodes that can directly communicate with it; i.e., $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$.

Define $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{N\times N}$ as the weight matrix associated to $\mathcal{G}$, with a_{ij} the weight associated to the exchange of information between nodes i and j. These weights satisfy the following assumption:

Assumption 2.1 (*Properties of the Weight Matrix*)**.** The weight matrix $\mathbf{A}$ is compatible with the underlying graph, $\mathcal{G}$, i.e., it is such that $a_{ii} > 0$, $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$, $a_{ij} > 0$ if and only if $(i, j) \in \mathcal{E}$, and satisfies $\mathbf{A1} = \mathbf{1}$, $\mathbf{1}^T\mathbf{A} = \mathbf{1}^T$, with $\mathbf{1} = (1, \ldots, 1)^T$.

Since the communication graph is connected, the assumption implies that $\mathbf{A}$ has one eigenvalue $\lambda_1 = 1$ with associated right eigenvector $\mathbf{1}$ and algebraic multiplicity equal to one. The rest of the eigenvalues, sorted in decreasing order, satisfy $1 > \lambda_2 \geq \cdots \geq \lambda_N > -1$. Without loss of generality, we assume that the algebraic connectivity is defined by λ_2, i.e., $|\lambda_2| \geq |\lambda_N|$. Note that the assumption on doubly stochastic weights can be easily satisfied using, e.g., Metropolis Weights, Xiao and Boyd (2004), or using distributed balancing techniques such as the ones proposed by Gharesifard and Cortés (2012) and Priolo, Gasparri, Montijano, and Sagues (2013).

Each node in the network is able to measure a local, continuous physical process, $r_i : \mathbb{R} \to \mathbb{R}$, $i \in \{1, \ldots, N\}$. Let $\mathbf{r}(t) = (r_1(t), \ldots, r_N(t))^T$ be the vector of the signals measured by each node. The final goal of the network is to design a distributed algorithm that enables anonymous nodes to eventually track the average of the signal inputs $r_i(t)$, $i \in \{1, \ldots, N\}$, using only local information. We denote this average by $\bar{r}(t)$,

$$\bar{r}(t) = \frac{1}{N}\sum_i r_i(t).$$

In order to compute $\bar{r}(t)$ each node maintains an estimation $x_i : \mathbb{N} \to \mathbb{R}$, $i \in \{1, \ldots, N\}$, which is updated at discrete times, $n \in \mathbb{N}$. The vector with all the estimations is denoted by $\mathbf{x}(n) = (x_1(n), \ldots, x_N(n))^T$. The sample period used by the nodes to estimate $\bar{r}(t)$ is denoted by h. Therefore, the relationship between the continuous time and the discrete time updates is defined by $nh = t$.

Another important issue that we will analyze in the paper is how big h can be while ensuring that $\mathbf{x}(n) \to \bar{r}(nh)\mathbf{1}$ with a sufficiently small error as n evolves. For the sake of brevity in the notation, along the paper we will omit the time dependence of the input signals, using $r_i(n)$ to denote the value of the input at time instant nh.

3. RKODAC: robust *k*-order dynamic average consensus algorithm

In this section, we propose a consensus algorithm to achieve the robust dynamic average consensus. We let $k \in \mathbb{N}$ be the order of the algorithm used to solve the problem, which is fixed a priori and equal for all the nodes in the network. From now on we will refer to this algorithm as the kth order dynamic consensus algorithm. We will show later the kind of signals that can be tracked by the method depending on the value of k.

Let us define the standard kth order differences by

$$\Delta^{[k]}r_i(n) = \Delta^{[k-1]}r_i(n) - \Delta^{[k-1]}r_i(n-1), \tag{1}$$

where $\Delta^{[0]}r_i(n) = r_i(n)$. We let $\Delta^{[k]}\bar{r}(n)$ represent the variation in the average of the kth difference of the signals, i.e.,

$$\Delta^{[k]}\bar{r}(n) = \frac{1}{N}\sum_i \Delta^{[k]}r_i(n). \tag{2}$$

In the kth order dynamic consensus each node exchanges with its neighbors a k-dimensional variable $(x_i^{[1]}, \ldots, x_i^{[k]})$. We denote by $\mathbf{x}^{[\ell]} = (x_1^{[\ell]}, \ldots, x_N^{[\ell]})^T$ the vector with the ℓth component of the variables of all the nodes. The update executed by the nodes has the following cascade form:

$$\begin{aligned} x_i^{[1]}(n+1) &= \gamma_n\left[a_{ii}x_i^{[1]}(n) + \sum_{j\in\mathcal{N}_i} a_{ij}x_j^{[1]}(n)\right] \\ &\quad + \Delta^{[k]}r_i(n) + (1-\gamma_n)\Delta^{[k-1]}r_i(n-1), \\ x_i^{[\ell]}(n+1) &= \gamma_n\left[a_{ii}x_i^{[\ell]}(n) + \sum_{j\in\mathcal{N}_i} a_{ij}x_j^{[\ell]}(n)\right] \\ &\quad + x_i^{[\ell-1]}(n+1), \end{aligned} \tag{3}$$

for $\ell \in \{2, \ldots, k\}$, where a_{ij} are the local weights, $i \in \{1, \ldots, N\}$, and $\gamma_n > 0$ are damping factors, which will be defined later in the section.

Let us note the distributed nature of the algorithm, in which the nodes, besides the damping factors, only use their signal and the estimations of the average provided by neighbors in the communication graph.

Denoting $\Delta^{[\ell]}\mathbf{r}(n) = (\Delta^{[\ell]}r_1(n), \ldots, \Delta^{[\ell]}r_N(n))^T$, the update can also be put in vectorial form by

$$\begin{aligned} &\mathbf{x}^{[1]}(n+1) = \gamma_n\mathbf{A}\mathbf{x}^{[1]}(n) + \Delta^{[k]}\mathbf{r}(n) + (1-\gamma_n)\Delta^{[k-1]}\mathbf{r}(n-1), \\ &\mathbf{x}^{[\ell]}(n+1) = \gamma_n\mathbf{A}\mathbf{x}^{[\ell]}(n) + \mathbf{x}^{[\ell-1]}(n+1), \end{aligned} \tag{4}$$

for $\ell \in \{2, \ldots, k\}$.

Remark 3.1. Intuitively, $\mathbf{x}^{[1]}(n+1)$ aims to be an approximation to $\Delta^{[k-1]}\bar{r}(n)$. The term $\mathbf{A}\mathbf{x}^{[1]}(n)$ contributes to improving $\mathbf{x}^{[1]}(n)$ as an approximation to the average of $(k-1)$th order differences at iteration $n-1$, $\Delta^{[k-1]}\bar{r}(n-1)$, whereas the term $\Delta^{[k]}\mathbf{r}(n)$ contributes to time-advancing the approximation from $(n-1)$ to n, that is, it acts as a correction term. The factor γ_n is introduced to damp the initial errors and the term $(1-\gamma_n)\Delta^{[k-1]}\mathbf{r}(n-1)$ is needed to compensate for the effect of γ_n on $\mathbf{A}\mathbf{x}^{[1]}(n)$. The successive terms, $\mathbf{x}^{[\ell]}(n+1)$, contain the estimation of $\Delta^{[k-\ell]}\bar{r}(n)$, which are computed by averaging the estimation at the previous time and summing the average estimation of the next order difference, $\mathbf{x}^{[\ell-1]}(n+1)$, to