Automatica 85 (2017) 441–447

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Perturbing consensus for complexity: A finite-time discrete biased min-consensus under time-delay and asynchronism[☆]

Yinyan Zhang, Shuai Li

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 3 December 2016
Received in revised form 26 April 2017
Accepted 24 July 2017
Available online 19 September 2017

Keywords:
Consensus
Min-consensus
Time-delay
Asynchronism
Maze solving
Shortest path planning
Complete coverage

ABSTRACT

In this paper, we propose a discrete-time biased min-consensus protocol with finite-time convergence by perturbing an existing min-consensus protocol, and investigate its convergence under time-delay and a synchronous state update. It is shown that a complex behavior that can address shortest path planning on a graph emerges from this modified consensus protocol. Theoretical analysis shows that the proposed protocol converges in finite time. In real-world networked systems, there may exist inevitable time delay or asynchronism in state updates. The convergence of biased min-consensus under these non-ideal situations is also theoretically analyzed. To show the scalability and efficiency of the proposed protocol, it is applied to large-scale maze solving on a maze map containing 640×640 pixels, which corresponds to a graph with 42,185 nodes. In addition, we also present an application of the proposed protocol to address the complete coverage problem, which further demonstrates the potential of biased min-consensus in robotic applications.

1. Introduction

Consensus includes min-consensus, max-consensus and average consensus (Cortés, 2008). Depending on the existence of a leader, consensus is divided into leader–follower consensus and leaderless consensus (Abaid & Porfiri, 2012; Olfati-Saber & Murray, 2004). Consensus protocols are divided into continuous-time consensus protocols (Olfati-Saber & Murray, 2004) and discrete-time consensus protocols (Cai & Ishii, 2012). Considering that, in practice, consensus protocols are generally implemented in a discrete-time manner, we consider discrete-time consensus protocols. Effort has been paid to the design of distributed discrete-time protocols to address time-delay (Xiao & Wang, 2008), state observation (Xu, Chen, Huang, & Gao, 2013), transmission nonlinearity (Chen, Lü, & Lin, 2013), and input saturation constraints (Yang, Meng, Dimarogonas, & Johansson, 2014). In terms of applications, discrete-time min-consensus protocols have been applied to network utility maximization (He, Duan, Hou, Cheng, & Chen, 2015), frequency regulation in islanded AC microgrids (Cady, Domíguez-García, & Hadjicostis, 2015), and determining when averaging consensus is reached (Yadav & Salapaka, 2007). Although many results have been reported on the design of distributed discrete-time consensus protocols and their applications, e.g., Cady et al. (2015), Cai and Ishii (2012), He et al. (2015) and Yadav and Salapaka (2007), it remains unknown whether discrete-time consensus protocols can be used or extended to solving shortest path problems (Fu, Sun, & Rilett, 2006).

In this paper, we propose a discrete-time biased min-consensus protocol which is capable of generating shortest path planning. A comparison of the resultant shortest path planning algorithm with some existing algorithms is shown in Table 1. Note that, compared with centralized algorithms, distributed ones are more suitable for large-scale problems (Qin, Yu, & Hirche, 2012). We also extend the proposed protocol to solving maze problems (Ni, He, Wen, & Xu, 2013) and complete coverage problems (Yang & Luo, 2004). Main contributions of this paper are summarized as below: (1) A discrete-time biased min-consensus protocol is proposed, which is capable of generating shortest path planning; (2) The proposed protocol is asymptotically stable under time-delay or asynchronous state update, by which the state values of nodes converge in finite time; (3) To our knowledge, this is the first time that maze solving and complete coverage are achieved by modifying a discrete-time consensus protocol.

2. Preliminary

The graph theory (Godsil & Royal, 2001) is a useful tool for investigations on consensus of network systems. Let $\mathbb{G} = (\mathbb{V}, \mathbb{E})$

☆ This work was supported by the National Natural Science Foundation of China (with number 61401385), by Hong Kong Research Grants Council Early Career Scheme (with number 25214015), and also by Departmental General Research Fund of Hong Kong Polytechnic University (with number G.61.37.UA7L). The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonis Papachristodoulou under the direction of Editor Christos G. Cassandras.

E-mail addresses: yinyan.zhang@connect.polyu.hk (Y. Zhang), shuaili@polyu.edu.hk (S. Li).

http://dx.doi.org/10.1016/j.automatica.2017.08.014
0005-1098/

Table 1
Comparisons of algorithms for solving shortest path problems.

	Type	Convergence	Destinations	Path
This paper	Distributed	Yes	Multiple[c]	Shortest
Dijkstra algorithm (Dijkstra, 1959)	Centralized	Yes	One[b]	Shortest
Genetic algorithm (Ahn & Ramakrishna, 2002)	Centralized	Yes	One[b]	[a]
PSO (Mohemmed, Sahoo, & Geok, 2008)	Centralized	Yes	One[b]	[a]
ID algorithm (Jan, Sun, Tsai, & Lin, 2014)	Centralized	Yes	One[b]	Near-shortest
CPCNN (Sang, Lv, Qu, & Yi, 2016)	Centralized	Yes	One[b]	[a]

[a] The algorithm usually, but not always, produces a shortest path.
[b] At each run, the algorithm, if successful, only finds a (near-) shortest path from a given initial position to a destination.
[c] At each run, the algorithm finds a shortest one among the paths from a given initial position to multiple destinations.

denote an undirected connected graph with $\mathbb{V} = \{1, 2, \ldots, n\}$ denoting the set of nodes and $\mathbb{E}$ denoting the set of arcs on the graph, respectively. The value of node i in the graph is denoted by x_i. The arc connecting node i and node j is denoted by $(i; j)$ with $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$, where n denotes the number of nodes in the graph. The set of neighbors of node i is denoted by $\mathbb{N}(i) = \{j \mid (i; j) \in \mathbb{E}\}$. The weight of arc $(i; j)$ in an undirected graph is denoted by w_{ij}. Specifically, if arc $(i; j)$ exists, then $w_{ij} = w_{ji} > 0$; otherwise, $w_{ij} = 0$. In weighted undirected connected graph $\mathbb{G}$, a walk is an alternating sequence of adjacent edges. The sum of the weights of the edges is called the length of the walk. A path is a walk without repeated edges or nodes.

Consider a network of n nodes defined on graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$. Let $t_0 < t_1 < t_2 < \cdots$ be the time instants when the state of the network undergoes change. Let x_i^k denote the value of node i at time t_k. The min-consensus is such that the network asymptotically achieves consensus with $\lim_{k\to+\infty} x_i^k = \min_{i\in\mathbb{V}}\{x_i^0\}$, $\forall i \in \mathbb{V}$, where x_i^0 denotes the initial state value of node i. Finite-time convergence means that convergence is reached in finite updates, i.e., finite time, which is better than asymptotic convergence (Wang & Xiao, 2010). In a general network system, there are two types of nodes, i.e., leader nodes and follower nodes. Let $\mathbb{S}$ and $\mathbb{V} - \mathbb{S}$ denote sets of leader nodes and follower nodes, respectively. For leader-following network systems, with static leaders, an intuitive distributed min-consensus protocol is

$$\begin{cases} x_i^{k+1} = x_i^k, \ i \in \mathbb{S}, \\ x_i^{k+1} = \min\limits_{j\in\mathbb{N}(i)}\{x_j^k\}, \ i \in \mathbb{V} - \mathbb{S}. \end{cases} \tag{1}$$

Definition 1 (*Godsil and Royal (2001)*)**.** The shortest path problem defined on graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is to find a path from a node $s \in \mathbb{V}$ to another node $v \in \mathbb{V}$ such that the sum of the weights of its constituent edges is minimized.

The shortest path problem becomes more complicated when there are multiple destinations nodes. In this case, one needs to find a shortest path among the paths from a node to multiple destination nodes.

3. Discrete-time biased min-consensus

3.1. Protocol

We consider such a case, where the information of neighbor $j \in \mathbb{N}(i)$ that follower node i receives is $x_j^k + w_{ij}$, i.e., there is a biased term w_{ij}. Under the synchronous update, adding the biased term to the min-consensus protocol (1) yields the following biased min-consensus protocol:

$$\begin{cases} x_i^{k+1} = x_i^k, \ i \in \mathbb{S}, \\ x_i^{k+1} = \min\limits_{j\in\mathbb{N}(i)}\{x_j^k + w_{ij}\} \ i \in \mathbb{V} - \mathbb{S}. \end{cases} \tag{2}$$

Evidently, the leader nodes are static nodes, for which, $x_i^k = x_i^0$, $\forall i \in \mathbb{S}$. This consensus protocol is synchronous in the sense that all the nodes in the network update their state values at the same time instant.

Remark 1. Biased min-consensus protocol (2) is distributed since each follower node updates its state value based on information from neighbor nodes and each leader node is static. Due to the existence of biased term w_{ij} in biased min-consensus protocol (2), the network cannot achieve min-consensus. With the proposed leader–follower biased min-consensus protocol, the state values of the follower nodes do not converge to a common value. We call the protocol a leader–follower one since the state values of the follower nodes at the steady-state depend on the leader nodes, which is theoretically analyzed later. As the protocol (2) is distributed, the computational complexity can be analyzed by considering the floating point operations (FLOPs) (Boyd & Vandenberghe, 2004; Fang & Chan, 2009) needed for each node at each state update. A floating-point operation is defined as one addition, subtraction, multiplication, or division of two floating-point numbers (FLOPs) (Boyd & Vandenberghe, 2004; Fang & Chan, 2009). According to Fang and Chan (2009), calculating the square root of a non-negative real number costs 6 FLOPs. The max function $\min(a, b)$ with a and b being real numbers can be calculated via $\min(a, b) = (a + b + \sqrt{(a - b)^2})/2$, which thus costs 11 FLOPs. At each state update, the leader nodes are static, and thus cost no FLOPs. Let $N_{\max}$ denote the maximal number of neighbors of a node in the network. Then, in view of (2), at each update, each follower node requires at most $11(N_{\max} - 1) + N_{\max} = 12N_{\max} - 11$ FLOPs. Note that $N_{\max}$ is generally limited. For example, for the maze solving and coverage problems considered in this paper, $N_{\max} = 8$. Thus, $N_{\max}$ can generally be viewed as a sufficiently large constant. Therefore, at each update, the computational complexity of each follower node with biased min-consensus protocol (2) is $O(1)$, and that of each leader node is 0.

Synchronous bias-min consensus protocol (2) with bounded time-delay is formulated as follows:

$$\begin{cases} x_i^{k+1} = x_i^k, \ i \in \mathbb{S}, \\ x_i^{k+1} = \min\limits_{j\in\mathbb{N}(i)}\{x_j^{k-\tau_{ij}} + w_{ij}\}, \ i \in \mathbb{V} - \mathbb{S}, \end{cases} \tag{3}$$

where $\tau_{ij} \in \{1, 2, 3, \ldots\}$ is the variable to scale the time-delay between node i and node j. In this paper, we set $x_i^0 = x_i^{-1} = x_i^{-2} = \cdots, \forall i \in \mathbb{V}$.

Synchronous communication requires a central synchronizing clock, which may not be available in practical applications (Fang & Antsaklis, 2005; Qin et al., 2012). Therefore, it is worth investigating consensus under asynchronous communication, where the state update of nodes is not synchronous. Let nonempty set $\mathbb{U}(k) \subset \mathbb{V}$ denote the set of updating indexes of nodes on graph $\mathbb{G}$ at the kth time instant. The terminology "time instant" here is independent for all the nodes, which is used only for the convenience of analysis and illustration. Based on min-consensus consensus protocol (1), a