Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Multi-Agent Consensus Using Both Current and Outdated States ⋆

Yongcan Cao * **Wei Ren** ** **YangQuan Chen** ***

* *Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA (Tel: 1-435-797-2845; e-mail: yongcan@cc.usu.edu)*
** *Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA (e-mail: wren@engineering.usu.edu)*
*** *Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA (e-mail: yqchen@engineering.usu.edu)*

Abstract: We propose a distributed consensus algorithm for multi-agent systems. In contrast to the standard consensus algorithm that relies on only current states, the proposed algorithm uses both current states and outdated states stored in memory. The proposed algorithm is analyzed under an undirected communication graph. It is shown that the proposed algorithm converges faster than the standard consensus algorithm while requiring identical maximum control effort if the outdated states are chosen properly. Simulation results demonstrate the effectiveness of the proposed algorithm.

Keywords: Consensus, Multi-agent systems, Cooperative control, Outdated states

1. INTRODUCTION

Autonomous vehicles have received significant attention due to their potential applications in both civilian and military sectors. The vehicles can replace human beings in various dangerous environments such as in hazardous chemical factories, deep sea, coal mining, etc. Even though autonomous vehicles can perform well when executing a solo task, multiple autonomous vehicles can perform even better through cooperation and coordination.

As a distributed strategy to multi-vehicle cooperative control, consensus algorithms have received significant attention recently (see Ren et al. [2007] and references therein). Consensus algorithms rely on neighbor-to-neighbor information exchange. Through updating the information states received from its neighbors, each vehicle responds to the deviations between its state and its neighbors' states. With proper communication among all vehicles, their information states will converge to a common value. Examples of information states include positions, velocities, orientations, and phases.

In Jadbabaie et al. [2003], a nearest neighbor rule is analyzed in a multi-agent consensus context under an undirected communication graph. Olfati et at. [2004] studies average consensus problems over a directed communication graph. In Moreau [2005] and Ren et al. [2005], the results in Jadbabaie et al. [2003] are extended to a directed communication graph. Recent research also includes extensions of consensus algorithms. For example, in Jin et al. [2006], the author considers multi-hop relay in consensus problems. The control input of each agent depends not only on its neighbors' states, but also on its neighbors' neighbors' states. By introducing more information with second hop, Jin et al. [2006] demonstrates that the consensus speed is improved. However, the tradeoff for introducing second-hop information is that extra communication and larger control effort are required.

In this paper, we introduce a distributed consensus algorithm that uses both current states and outdated states stored in memory. The motivation is that 1) outdated state information is within any control system and deserves consideration, and 2) memory is very cheap. In contrast, the standard consensus algorithm (see Ren et al. [2007] and references therein) relies on only current states. While there is a vast literature on consensus algorithms with delays (see references surveyed in Ren et al. [2007]), the delays are considered a negative factor and the focuses there are usually on how the delays affect the stability of the consensus algorithms. Compared with the algorithm in Jin et al. [2006], our algorithm does not require second hop communication. The difference between this paper and other work in time-delay systems (see Chen et al. [1994], Olgac et al. [2002], Gu et al. [2003]) is that outdated state information is considered as a positive factor and is, therefore, applied to multi-agent consensus problems while Chen et al. [1994], Olgac et al. [2002], Gu et al. [2003] consider the effect of time delay on stability of a single system. We will show that with both current states and extra outdated states stored in memory, the proposed algorithm converges faster than the standard consensus algorithm while requiring identical maximum control effort when the outdated states are chosen properly.

The remainder of this paper is organized as follows. In Section 2, we introduce definitions and background. In Section 3, we introduce a distributed consensus algorithm using both current and outdated states and provide convergence analysis. Simulation results are given in Section 4 while Section 5 contains the conclusion.

⋆ This work was supported in part by a National Science Foundation CAREER Award (ECCS-0748287), the Utah Water Research Laboratory, and the Community/University Research Initiative.

2. DEFINITIONS AND BACKGROUND

Suppose that there are n agents in a team. We model information exchange among the n agents by an undirected graph $\mathcal{G} = (V, W)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $W \subseteq V^2$ represent the node set and the edge set, respectively. The edge $(v_i, v_j) \in W$ denotes that both agent i and agent j can access each other's information. All the neighbors of agent i are denoted as $N(i) = \{v_j | (v_j, v_i) \in W$. A path is a sequence of edges of the form (v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), ... Graph $\mathcal{G}$ is connected if there is a path between every pair of distinct agents.

The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ associated with graph $\mathcal{G}$ is defined as $a_{ij} > 0$ if $(v_j, v_i) \in W$ and $a_{ij} = 0$ otherwise. Because $\mathcal{G}$ is undirected, A is symmetrical. The Laplacian matrix $L = [\ell_{ij}] \in \mathbb{R}^{n\times n}$ associated with $\mathcal{G}$ is defined as

$$\ell_{ij} = \begin{cases} \sum_{j\in N(i)} a_{ij}, & i = j, \\ -a_{ij}, & i \neq j. \end{cases} \tag{1}$$

Because $\mathcal{G}$ is undirected, L is symmetric positive semidefinite and has a zero eigenvalue with an eigenvector $\mathbf{1}_n \in \mathbb{R}^n$, where $\mathbf{1}_n = [1, 1, \ldots, 1]^T$. If $\mathcal{G}$ is connected, then L has a simple zero eigenvalue and all the other eigenvalues are positive (see Chung [1997]).

In this paper, we will consider agents with single-integrator dynamics given by

$$\dot{\xi}_i(t) = u_i(t), \quad i = 1, \ldots, n, \tag{2}$$

where $\xi_i(t) \in \mathbb{R}$ and $u_i(t) \in \mathbb{R}$ represent, respectively, the state of the ith agent and the associated control input. A standard consensus algorithm for (2) is

$$u_i(t) = -\sum_{j\in N(i)} a_{ij}[\xi_i(t) - \xi_j(t)], \tag{3}$$

where a_{ij} is the (i, j) entry of adjacency matrix A.

Using (3), (2) can be written in matrix form as

$$\dot{\Xi}(t) = -L\Xi(t),$$

where $\Xi(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_n(t)]^T \in \mathbb{R}^n$ and L is the corresponding Laplacian matrix defined in (1).

Consensus is reached among the n agents if for all $\xi_i(0)$, $\xi_i(t) \to \xi_j(t)$ as $t \to \infty$. Using (3), consensus is reached if and only if graph $\mathcal{G}$ is connected (equivalently, L has a simple zero eigenvalue and all the other eigenvalues are positive).

3. CONSENSUS ALGORITHM USING BOTH CURRENT AND OUTDATED STATES

In this section, we will introduce and analyze a consensus algorithm using both current and outdated states. We will focus on how this algorithm can affect the performance, especially the convergence speed and the maximum control effort, in contrast to the standard algorithm (3).

We propose a consensus algorithm that uses both current and outdated states as

$$u_i(t) = -\sum_{j\in N(i)} a_{ij}\{[\xi_i(t) - \xi_j(t)] + [\xi_i(t-\tau) - \xi_j(t-\tau)]\}, \tag{4}$$

where $\xi_i(t)$ is the current state of the ith agent and $\xi_i(t-\tau)$ is the outdated state stored in memory, where $\tau > 0$ characterizes how old the state is. The larger τ is, the more outdated $\xi_i(t-\tau)$ is. In the following, we assume that $\xi_i(t-\tau) \triangleq \xi_i(t)$ for $0 \le t < \tau$. In contrast to (3), (4) introduces extra outdated states stored in memory.

To form a common basis for comparison, we will compare (4) with

$$u_i(t) = -\sum_{j\in N(i)} a_{ij}\{[\xi_i(t) - \xi_j(t)] + [\xi_i(t) - \xi_j(t)]\}. \tag{5}$$

In essence, we have replaced one term $\xi_i(t) - \xi_j(t)$ in (5) with $\xi_i(t-\tau) - \xi_j(t-\tau)$ to get (4).

3.1 Single-agent Case

In this subsection, we focus on the single-agent case. The results here will serve as a basis for analysis in the multi-agent case. We consider a single agent system with single-integrator dynamics given by

$$\dot{\phi}(t) = w(t), \tag{6}$$

where $\phi \in \mathbb{R}$ is the state and $w \in \mathbb{R}$ is the control input, and compare the following two control inputs:

$$w(t) = 2a\phi(t), \tag{7}$$

$$w(t) = a\phi(t) + a\phi(t-\tau), \tag{8}$$

where $a < 0$ and $\tau > 0$. We assume that $\phi(t-\tau) \triangleq \phi(t)$ for $0 \le t < \tau$. The difference between (7) and (8) is that $\phi(t)$ in (7) is partly replaced with $\phi(t-\tau)$ in (8). In other words, (8) uses both current and outdated states while (7) uses only the current state. If (8) achieves larger convergence speed and requires no larger control effort than (7), we say that (8) achieves better performance than (7). As shown below, while (8) cannot outperform (7) for any $\tau > 0$, (8) outperforms (7) when τ is chosen properly.

Before moving on, we need the following lemma.

Lemma 3.1. All closed-loop poles of system (6) using control input (8) are on the open left half plane if $\tau > 0$. In addition, the real parts of all these poles are smaller than $2a$ if τ satisfies $\tau \in (0, \zeta)$, where ζ is the minimal positive real number satisfying $\cos a\zeta\sqrt{e^{-4a\zeta} - 1} = e^{2a\zeta}$.

Proof: The Laplace transform of system (6) using (8) can be written as

$$s\phi(s) - \phi(0) = a\phi(s) + ae^{-s\tau}\phi(s),$$

where $\phi(s)$ is the Laplace transform of $\phi(t)$ and $\phi(0)$ is the initial condition of $\phi(t)$. The closed-loop poles therefore satisfy

$$s - (1 + e^{-s\tau})a = 0. \tag{9}$$

Let $s = x + yi$, where $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are, respectively, the real and imaginary parts of s, then we can get

$$x = a[1 + e^{-\tau x}\cos(y\tau)], \tag{10}$$

$$y = -ae^{-\tau x}\sin(y\tau). \tag{11}$$

From (10) and (11), we have

$$(x - a)^2 + y^2 = a^2e^{-2\tau x}. \tag{12}$$

Suppose that $x > 0$, then it follows that $|e^{-\tau x}\cos(y\tau)| < 1$, which implies $1 + e^{-\tau x}\cos(y\tau) > 0$. Then it follows from (10) that $x = a[1 + e^{-\tau x}\cos(y\tau)] < 0$ because $a < 0$, which results in a contradiction. Similarly, suppose that $x = 0$, then we have $y = 0$ according to (12). Therefore, it follows from (10) that $x = a[1 + e^{-\tau x}\cos(y\tau)] = 2a < 0$, which also results in contradiction. Therefore, we conclude that $x < 0$ for all $\tau > 0$.

Note that (9) [1] has the property that each root is continuous with respect to τ. Also note that

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}\tau}|_{\tau=0} &= -ae^{-\tau x}(\tau\frac{\mathrm{d}x}{\mathrm{d}\tau}+x)\cos(y\tau)|_{\tau=0}\\ &\quad -ae^{-\tau x}(y+\tau\frac{\mathrm{d}y}{\mathrm{d}\tau})\sin(y\tau)|_{\tau=0}\\ &= -ax<0,\end{aligned}$$

where we have used the fact that $x<0$ from the first statement of the lemma. The continuity of $\frac{\mathrm{d}x}{\mathrm{d}\tau}$ with respect to τ implies that there exists $\varepsilon>0$ such that $\frac{\mathrm{d}x}{\mathrm{d}\tau}\leq 0$ for all $0<\tau\leq\varepsilon$. Note that $x=2a$ when $\tau=0$. Also note that x is a continuous function of τ, it follows that $x<2a$ when $0<\tau<\zeta$, where ζ is the minimal positive real number such that $x(\tau)|_{\tau=\zeta}=2a$. Equivalently, ζ is the minimal positive real number satisfying

$$2a = a[1+e^{-2a\zeta}\cos(y\zeta)], \tag{13}$$

$$y = -ae^{-2a\zeta}\sin(y\zeta). \tag{14}$$

From (13), we have $\cos(y\zeta)=e^{2a\zeta}$, then (14) becomes

$$y = \pm ae^{-2a\zeta}\sqrt{1-e^{4a\zeta}}. \tag{15}$$

Combining (13) and (15), we get $\cos a\zeta\sqrt{e^{-4a\zeta}-1}=e^{2a\zeta}$. ∎

Note that the closed-loop pole of system (6) using (7) is $2a$. From Lemma 3.1, we can see that the closed-loop poles of system (6) using (8) are always on the left hand side of the closed-loop pole of system (6) using (7) when $\tau\in(0,\zeta)$. As a result, system (6) using control input (8) converges faster than system (6) using control input (7). The following lemma will show that the maximal control effort using (8) is identical to that using (7).

Lemma 3.2. The maximum control effort using (8) is identical to that using (7). In particular, $\max_{t\geq 0}|w(t)| = |w(0)| = 2|a\phi(0)|$.

Proof: For system (6) using control input (7), we know that the state $|\phi(t)|$ exponentially decays because the only closed-loop pole is $2a<0$, which implies that $\max_{t\geq0}|\phi(t)|=|\phi(0)|$. Therefore, it follows that $\max_{t\geq0}|w(t)|=\max_{t\geq0}2|a\phi(t)|=2|a\phi(0)|$.

For system (6) using control input (8), there are infinity number of closed-loop poles. However, all poles are on the open left half plane when $\tau>0$ as shown in Lemma 3.1. Therefore, the state will decay with possible oscillations. For $0\leq t<\tau$, we get $w(t)=2a\phi(t)$ because $\phi(t-\tau)\triangleq\phi(t)$ for $0\leq t<\tau$. Therefore, $|\phi(t)|$ will exponentially decay, which implies that $|w(t)|=2|a\phi(t)|\leq 2|a\phi(0)|$ for $0\leq t<\tau$. For $t=\tau$, we have $\phi(t-\tau)=\phi(0)$, which implies that $|w(\tau)|=|a\phi(\tau)+a\phi(0)|\leq|a\phi(\tau)|+|a\phi(0)|\leq 2|a\phi(0)|$. For $t>\tau$, because all the closed-loop poles of system (6) using (8) are on the open left half plane, it follows that $|\phi(t)|<|\phi(\tau)|$ and $|\phi(t-\tau)|<|\phi(0)|$ for $t>\tau$, which implies that $|w(t)|<|w(\tau)|$ for $t>\tau$. Combing the above arguments, we can see that $\max_{t\geq0}|w(t)|=2|a\phi(0)|$ using (8). ∎

To illustrate, we compare (7) with (8). Let $a=-1$ and $\tau=0.25$ s in. Also let $\phi(0)=4$. The states and control inputs using (7) and (8) are shown in Figs. 1 and 2, respectively. From these two figures, we can see that the state using (8) converges to zero faster than the state using (7). In addition, we can see that using either (7) or (8), $\max_{t\geq0}|w(t)|=2|a\phi(0)|=8$.

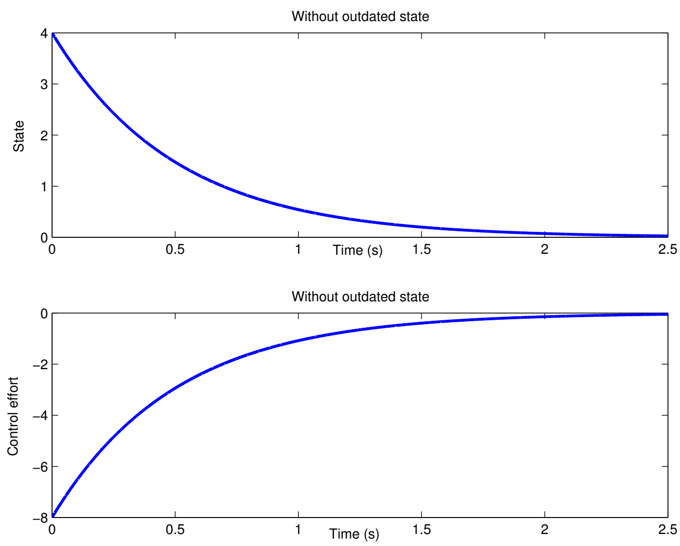

Fig. 1. States and control effort using (7)

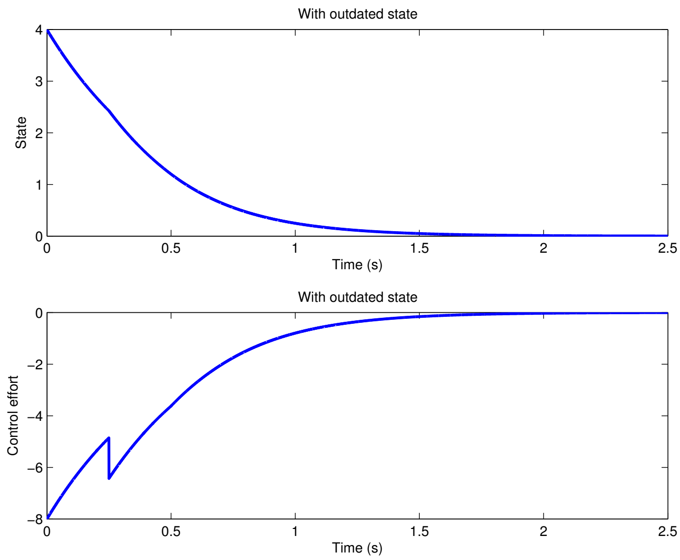

Fig. 2. States and control effort using (8)

From Lemmas 3.1 and 3.2, we can see that the convergence speed using (8) is larger than that using (7) while the maximal control effort remains unchanged. By utilizing both outdated and current state information, we can achieve better performance.

3.2 Multi-agent Case

In this subsection, we will analyze consensus algorithm (4) under an undirected communication graph and compare algorithm (4) with algorithm (5) in terms of convergence speed and maximum control effort.

Before moving on, we need the following lemma:

Lemma 3.3. Kim et al. [2006] Suppose that the undirected communication graph is connected. Let λ_i be the ith eigenvalue of L, where L is the Laplacian matrix defined in (1), where $\lambda_n\geq\lambda_{n-1}\geq\cdots\geq\lambda_2>\lambda_1=0$. The convergence speed (i.e. how fast consensus is reached) of algorithm (3) (respectively, algorithm (5)) is determined by λ_2 (respectively, $2\lambda_2$).

[1] Equation (9) is called quasipolynomial.

Theorem 3.1. Suppose that the undirected communication graph is connected. Let λ_i be the ith eigenvalue of L as in Lemma 3.3. Using (4), consensus is reached for any $\tau > 0$. In addition, algorithm (4) reaches consensus faster than algorithm (5) if τ satisfies $\tau \in (0, \zeta)$, where $\zeta = \min_{i=2,\ldots,n} \zeta_i$, where ζ_i, $i = 2, \ldots, n$, is the minimal positive scalar satisfying $\cos \lambda_i \zeta_i \sqrt{e^{4\lambda_i \zeta_i} - 1} = e^{-2\lambda_i \zeta_i}$.

Proof: Using (4), (2) can be rewritten in matrix form as

$$\dot{\Xi}(t) = -L\Xi(t) - L\Xi(t - \tau), \tag{16}$$

where $\Xi(t) = [\xi_1(t), \xi_2(t), \ldots, \xi_n(t)]^T \in I\!R^n$ and $L \in I\!R^{n \times n}$ is the corresponding Laplacian matrix. The Laplace transform of (16) is

$$s\Xi(s) - \Xi(0) = -L\Xi(s) - e^{-s\tau} L\Xi(s),$$

which can be rewritten as

$$(sI_n + L + e^{-s\tau} L)\Xi(s) = \Xi(0), \tag{17}$$

where $\Xi(0) = [\xi_1(0), \xi_2(0), \ldots, \xi_n(0)]^T \in I\!R^n$, and $I_n \in I\!R^{n \times n}$ is the identity matrix. The closed-loop poles of (2) using (4) satisfy

$$\det[sI_n + (1 + e^{-s\tau})L] = 0. \tag{18}$$

Noting that $-\lambda_i$ denotes the ith eigenvalue of $-L$, it follows that

$$\det(sI_n + L) = \prod_{i=1}^{n} (s + \lambda_i). \tag{19}$$

By comparing (18) and (19), we know that the closed-loop poles of (2) using (4) satisfy

$$\det[sI_n + (1 + e^{-s\tau})L] = \prod_{i=1}^{n} [s + (1 + e^{-s\tau})\lambda_i] = 0.$$

As a result, the closed-loop poles of (2) using (4) satisfy

$$s + (1 + e^{-s\tau})\lambda_i = 0, \quad i = 1, \ldots, n. \tag{20}$$

There may exist multiple roots of (20) corresponding to each λ_i. Noting that $\lambda_1 = 0$, it follows that the unique root of (20) corresponding to λ_1 is zero. Note that $-\lambda_i < 0$, $i = 2, \ldots, n$. According to Lemma 3.1 with $-\lambda_i$ playing the role of a, it follows that the roots of (20) corresponding to λ_i, $i = 2, \ldots, n$, are on the open left half plane. Thus (20) has a simple root at zero and all the other roots are on the open left half plane. Noting that $L\mathbf{1}_n = 0$, it follows that $\xi_i = \xi_j$ is an equilibrium of (2) using (4). Therefore, it follows that $\xi_i(t) \to \xi_j(t)$ as $t \to \infty$. That is, consensus is reached using (4) for any $\tau > 0$.

Similarly, according to Lemma 3.1 with $-\lambda_i$ playing the role of a, if $\tau \in (0, \zeta_i)$, $i = 2, \ldots, n$, then the real parts of all roots of (20) corresponding to λ_i, $i = 2, \ldots, n$, are smaller than $-2\lambda_i$, $i = 2, \ldots, n$. Therefore, if $\tau \in (0, \zeta)$, where $\zeta = \min_{i=2,\ldots,n} \zeta_i$, then the real parts of all roots of (20) are smaller than $-2\lambda_2$. Noting that the convergence speed of (2) using (5) is determined by $2\lambda_2$ according to Lemma 3.3, we conclude that algorithm (4) converges faster than algorithm (5). ■

Similar to Lemma 3.2, we have the following lemma for maximum control effort.

Lemma 3.4. The maximum control effort using (4) is identical to that using (5). In particular, $\max_i \max_{t \geq 0} |u_i(t)| = \max_i |u_i(0)|$.

Proof: Differentiating (5), gives

$$\dot{u}_i(t) = -\sum_{j \in N(i)} 2a_{ij}(\dot{\xi}_i(t) - \dot{\xi}_j(t)),$$

which can be written as

$$\dot{u}_i(t) = -\sum_{j \in N(i)} 2a_{ij}(u_i(t) - u_j(t)). \tag{21}$$

Equation (21) can be written in matrix form as

$$\dot{U}(t) = -2LU(t), \tag{22}$$

where $U(t) = [u_1(t), u_2(t), \cdots, u_n(t)]^T \in I\!R^n$. The solution to (22) is given as $U(t) = e^{-2Lt}U(0)$, where e^{-2Lt} is a row-stochastic matrix (see Ren et al. [2005]). It thus follows that $\|U(t)\|_\infty \leq \left\|e^{-2Lt}\right\|_\infty \|U(0)\|_\infty = \|U(0)\|_\infty$, where we have used the fact that the infinity norm of a row-stochastic matrix is one.

Similarly, differentiating (4), gives

$$\dot{u}_i(t) = -\sum_{j \in N(i)} a_{ij}\{[u_i(t) - u_j(t)] + [u_i(t - \tau) - u_j(t - \tau)]\},$$

which can be written in matrix form as

$$\dot{U}(t) = -LU(t) - LU(t - \tau). \tag{23}$$

Note that $U(t - \tau) \triangleq U(t)$ for $0 \leq t < \tau$. Thus (23) becomes $\dot{U}(t) = -2LU(t)$ for $0 \leq t < \tau$, which implies that $\|U(t)\|_\infty \leq \|U(0)\|_\infty$ for $0 \leq t < \tau$. When $t = \tau$, $u_i(\tau) = -\sum_{j \in N(i)} a_{ij}\{[\xi_i(\tau) - \xi_j(\tau)] + [\xi_i(0) - \xi_j(0)]\}$, which implies that $U(\tau) = \frac{1}{2}[\lim_{t \to \tau} U(t) + U(0)]$. Therefore, it follows that $\|U(\tau)\|_\infty = \frac{1}{2}\|\lim_{t \to \tau} U(t) + U(0)\|_\infty \leq \frac{1}{2}\|\lim_{t \to \tau} U(t)\|_\infty + \frac{1}{2}\|U(0)\|_\infty \leq \|U(0)\|_\infty$. Noting that consensus is achieved using (4), we conclude that $\|U(t)\|_\infty \leq \|U(\tau)\|_\infty$ when $t > \tau$. Therefore, it follows that $\|U(t)\|_\infty \leq \|U(0)\|_\infty$ for $t \geq 0$, i.e., $\max_i \max_{t \geq 0} |u_i(t)| = \max_i |u_i(0)|$. ■

From Lemma 3.4 and Theorem 3.1, we can see that using outdated states, algorithm (4) reaches consensus faster than algorithm (5) while requiring identical maximum control effort if the outdated states are chosen properly. The above result demonstrates that outdated states are meaningful if they can be used properly. Moreover, algorithm (4) proposed in the paper is quite simple with the only requirement that τ be determined beforehand.

4. SIMULATION RESULTS

In this section, we compare (4) and (5) in simulation. We consider a team of four agents. The communication topology among the four agents is shown in Fig. 3. The corresponding Laplacian matrix is chosen as

$$L = \begin{bmatrix} 1.8 & -0.6 & -0.6 & -0.6 \\ -0.6 & 1.8 & -0.6 & -0.6 \\ -0.6 & -0.6 & 1.8 & -0.6 \\ -0.6 & -0.6 & -0.6 & 1.8 \end{bmatrix}.$$

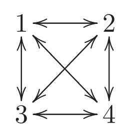

Fig. 3. Multi-agent undirected communication topology.

We let $\xi_i(0) = 6 - 2i$, $i = 1, \ldots, 4$. Figs. 4, 5, and 6 show, respectively, the results using (4) with small $\tau = 0.12$ s, (5), and (4) with large $\tau = 0.8$ s. We can see that consensus is achieved faster using (4) with $\tau = 0.12$ s than using (5) while

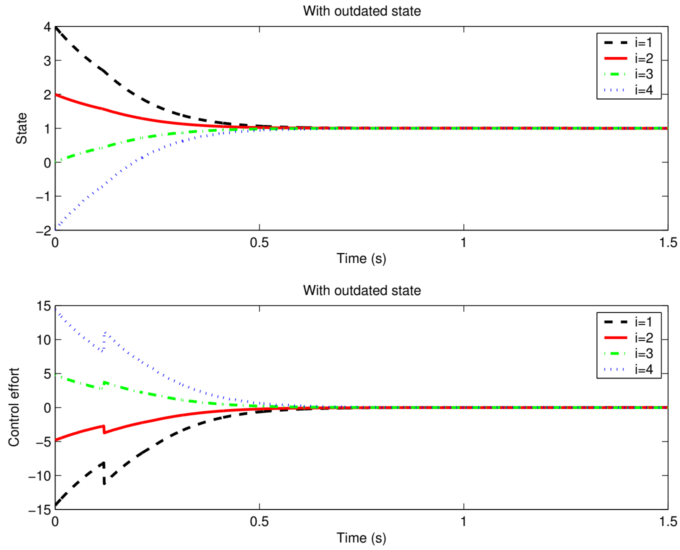

Fig. 4. States and control inputs using (4) with $\tau = 0.12$ s.

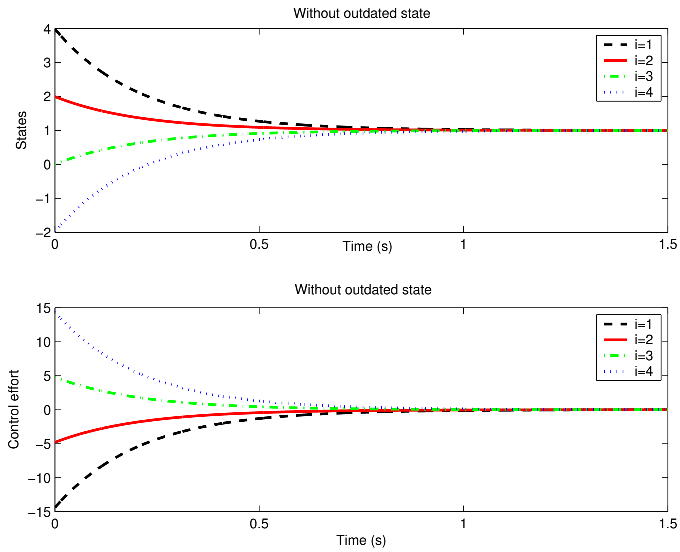

Fig. 5. States and control inputs using (5).

Table 1. Comparison between convergence time

Algorithm	within 5% of final equilibrium	within 2% of final equilibrium
(4) with $\tau = 0.12$ s	0.53 s	0.61 s
(5)	0.85 s	1.04 s
(4) with $\tau = 0.8$ s	7.12 s	9.36 s

consensus is achieved more slowly using (4) with $\tau = 0.8$ s than using (5). The maximum control effort is 14.4 in all cases.

Table 1 shows the convergence times using (4) with $\tau = 0.12$ s, (5), and (4) with $\tau = 0.8$ s. We can see that using both current and outdated states improves the convergence speed for consensus convergence if τ is below a certain bound defined explicitly in Theorem 3.1. However, when the delay is too large, using both current and outdated states has an adverse effect on the converge speed.

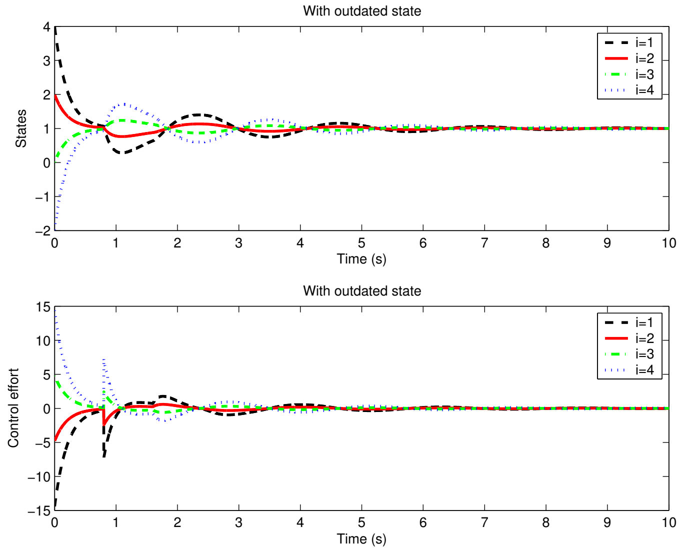

Fig. 6. States and control inputs using (4) with $\tau = 0.8$ s.

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a distributed consensus algorithm utilizing both current and outdated states. We have shown that the proposed consensus algorithm improves the convergence speed for consensus convergence without increasing the maximal control effort if the outdated states are chosen properly. Simulation results have shown the effectiveness of the proposed algorithm. Future work includes finding optimum outdated states and considering agents with dynamic models. In addition, we will consider an algorithm

$$u_i(t) = -\sum_{j\in N(i)} a_{ij}\{[\xi_i(t) - \xi_j(t-\sigma_j)] + [\xi_i(t-\tau) - \xi_j(t-\sigma_j-\tau)]\},$$

where σ_j denotes the delay caused by transmitting information from agent j to agent i and τ still represents how outdated the state is.

REFERENCES

J. Chen, G. Gu and C. N. Nett. A new method for computing delay margins for stability of linear delay systems. *Proc. of 33rd IEEE Conference on Decision and Control*, 433-437, Lake Buena Vista, FL, 2003.

F. R. K. Chung. Spectral graph theory. in *Regional Conference Series in Mathematics*, American Mathematical Society, 1997.

K. Gu, V. L. Kharitonov and J. Chen. *Stability of time-delay systems*. Birkhäuser, Basel, 2003.

A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48:988–1001, 2003.

Z. Jin and R. M. Murray. Multi-hop relay protocols for fast consensus seeking. *Proc. of 45th IEEE Conference on Decision and Control*, 1001–1006, San Diego, 2006.

Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. on Automatic Control*, 51:116–120, 2006.

L. Moreau. Stability of multi-agent systems with time-dependent communication links. *IEEE Trans. on Automatic Control*, 50:169–182, 2005.

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

N. Olgac and R. Sipahi. An exact method for the stability analysis of time-delayed linear time-invariant(LTI) systems. *IEEE Trans. on Automatic Control*, 47:793–797, 2002.

W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. on Automatic Control*, 50:655–661, 2005.

W. Ren, R. W. Beard and E. Atkins. Information Consensus in Multivehicle Cooperative Control: Collective Group Behavior through Local Interaction. *IEEE Control Systems Magazine*, 27:71–82, 2007.