

A Delayed Consensus Algorithm in Networks of Anticipatory Agents

Fatihcan M. Atay¹ and Dina Irofti²

Abstract—We introduce and analyze a delayed consensus algorithm as a model for interacting agents using anticipation of their neighbors’ states to improve convergence to consensus. We derive a necessary and sufficient condition for the system to reach consensus. Furthermore, we explicitly calculate the dominant characteristic root of the consensus problem as a measure of the speed of convergence. The results show that the anticipatory algorithm can improve the speed of consensus, especially in networks with poor connectivity. Hence, anticipation can improve performance in networks if the delay parameter is chosen judiciously, otherwise the system might diverge as agents try to anticipate too aggressively into the future.

I. INTRODUCTION

Consensus problems arise in a wide range of applications in distributed computing [9], management science [5], flocking and swarming theory [16], distributed control [6], and sensor networks [12], among others. In these applications, multi-agent systems interact to agree on a common value for a certain quantity of interest. The interaction rule that specifies the information exchange between the agents is called the *consensus protocol* or *consensus algorithm*. The consensus problem on networks in continuous time can be formulated as

$$\dot{x}_i = u_i(t), \quad i = 1, \dots, n, \quad (1)$$

where n is the number of agents in the network, $x_i \in \mathbb{R}$ is the state of the agent i at time t , which changes under the interaction with other agents as described by the consensus protocol $u_i(t)$. The system (1) is said to *reach consensus* if for any set of initial conditions $\{x_i(0)\}$ there exists $c \in \mathbb{R}$ such that $\lim_{t \rightarrow \infty} x_i(t) = c$ for all i , in which case the number c is called the *consensus value*.

In the classical linear case, the consensus protocol typically has the form

$$u_i(t) \sim \sum_{j=1}^n a_{ij}(x_j(t) - x_i(t)). \quad (2)$$

where a_{ij} are nonnegative numbers describing the pairwise interaction strength between agents. Under mild conditions related to the connectivity of the network, it can be shown that the system (1) under (2) reaches consensus from arbitrary initial conditions. A more interesting case is when the consensus protocol involves a time delay, which can

come in various forms. A well-known example is the delayed consensus protocol

$$u_i(t) \sim \sum_{j=1}^n a_{ij}(x_j(t - \tau) - x_i(t - \tau)) \quad (3)$$

which has been studied in [13]. One can view (3) as modeling *delayed information processing*, since the protocol feeds back the same information $\sum(x_j - x_i)$ to the system, as in (2), but only after a delay $\tau \geq 0$. In this case, it is known that there exists an upper limit τ_{\max} such that the system (1) under the protocol (3) reaches consensus from arbitrary initial conditions if and only if $\tau < \tau_{\max}$ [13]. Another type of interaction, which models *delayed information transmission*, is given by the consensus protocol [1], [2], [10], [14]

$$u_i(t) \sim \sum_{j=1}^n a_{ij}(x_j(t - \tau) - x_i(t)), \quad (4)$$

Here the interpretation is that the information coming from a neighbouring node j takes some time τ to reach site i . It was shown that the system (1) under the protocol (4) reaches consensus from arbitrary initial conditions regardless of the value of the delay τ as long as the network contains a spanning tree [1], [2].

In this paper we study another delayed consensus protocol where the delay comes from a quite different source, namely from the *anticipatory nature* of the agents. More precisely, we consider a network of *intelligent agents* who try to anticipate the future states of their neighbors in their interaction, which is a common situation in, e.g., economic systems. We therefore consider an anticipatory algorithm of the form

$$u_i(t) \sim \sum_{j=1}^n a_{ij}(\hat{x}_j(t + \tau) - x_i(t)), \quad (5)$$

where $\hat{x}_j(t + \tau)$ is the anticipated state of the neighbor x_j at some time $(t + \tau)$ in the future. In order to estimate the future state, the agents use a first order estimation derived from past observations, namely they employ *memory*. It is reasonable to expect that the memory window is about the same order of magnitude as the view into future, which will be assumed in this paper. Therefore, agent i , knowing the current state of $x_j(t)$ of a generic neighbor j and remembering its past state $x_j(t - \tau)$ as well, uses a *first-order estimation* to get the future state $\hat{x}_j(t + \tau)$ by linear extrapolation:

$$\begin{aligned} \hat{x}_j(t + \tau) &= x_j(t) + \frac{x_j(t) - x_j(t - \tau)}{\tau} \tau, \\ &= 2x_j(t) - x_j(t - \tau). \end{aligned} \quad (6)$$

¹Fatihcan M. Atay is with the Department of Mathematics, Bilkent University, 06800 Ankara, Turkey atay@member.ams.org

²Dina Irofti is with the Laboratoire des Signaux et Systèmes (L2S, UMR CNRS 8506), CNRS-CentraleSupélec-University Paris-Sud, 3, rue Joliot Curie, 91190, Gif-sur-Yvette, France Dina.Irofti@l2s.centralesupelec.fr

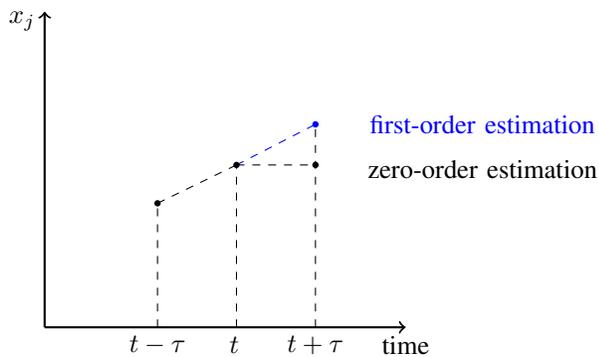


Fig. 1. Estimating the future state $x_j(t+\tau)$ of an agent j from its present and past states.

The idea is graphically summarized in Figure 1. By comparison, the classical consensus algorithm (2) can be viewed as a *zero-order estimation* where an agent's expectation of the short-term future of its neighbors is represented simply by the present states, $\hat{x}_j(t+\tau) = x_j(t)$.

Substituting (6) into (5), we thus obtain the consensus algorithm

$$u_i(t) \sim \sum_{j=1}^n a_{ij}(2x_j(t) - x_j(t-\tau) - x_i(t)), \quad (7)$$

which will be the focus of our study in this paper. In precise terms, we will study the convergence of the following consensus problem

$$\dot{x}_i(t) = \frac{1}{d_i} \sum_{j=1}^n a_{ij}(2x_j(t) - x_j(t-\tau) - x_i(t)), \quad (8)$$

where $a_{ij} = a_{ji} \forall i, j$ and $d_i = \sum_{j=1}^n a_{ij}$ is the (generalized) degree of node i . Dividing the summation by the degrees d_i gives rise to a *normalized Laplace operator*, which is a natural choice in several applications and has some advantageous properties, as will be briefly reviewed in Section II. In particular, the normalization bounds the spectrum of Laplacian regardless of the network size, thus allowing comparison of networks of very different sizes.

We note that τ in (8) is to be seen as a *design parameter*, rather than a delay imposed by system constraints as in (3) or (4). Nevertheless, (8) is still a delay-differential equation and its analysis is subject to the same difficulties one faces in studying infinite-dimensional systems on Banach spaces. The purpose of this paper is to investigate if and under what conditions the system (8) can reach consensus from arbitrary initial states, and furthermore, whether the introduction of a positive τ , namely *anticipation*, really brings any advantages into the consensus dynamics.

The paper is organized as follows. In Section II we introduce the basic notation and review relevant background information from graph theory. Section III presents the main results by giving necessary and sufficient conditions: we show that (8) reaches consensus from arbitrary initial states

if and only if $\tau < 1$. In Section IV we present a graphical depiction of the dominant roots for arbitrary eigenmodes of the system as a function of the delay value τ , thereby deriving a universal picture applicable for any undirected graph. In Section V we give simulation results from various networks and show that a positive τ can indeed improve speed of consensus, in some cases dramatically, especially in networks with poor connectivity. Hence, anticipation can improve performance in networks of interacting agents if the parameter τ is chosen judiciously, otherwise the system can diverge as agents try to anticipate too aggressively into the future.

II. PRELIMINARIES

We briefly review some relevant notions and notations from graph theory. For details, the reader is referred to standard texts such as [3] or [7].

A graph $G = (V, E)$ consists of a finite set V of vertices and a set of edges $E \subset V \times V$ consisting of unordered pairs of vertices. Two vertices i and j are called *neighbors* if $(i, j) \in E$. We consider simple, non-trivial, and undirected graphs without self-loops or multiple edges. We denote by $A = [a_{ij}]$ the (weighted) adjacency matrix of the graph, where $a_{ij} = a_{ji} > 0$ if i and j are neighbors, and $a_{ij} = 0$ otherwise. The degree d_i of node i is defined as $d_i = \sum_{j=1}^n a_{ij}$, i.e., the sum of the elements of the i^{th} row of A , and $D = \text{diag}(d_1, \dots, d_n)$ denotes the diagonal degree matrix.

The normalized Laplacian matrix is defined as

$$L = I_n - D^{-1}A, \quad (9)$$

where n is the number of nodes in the network, I_n is the identity matrix of size n . The normalized Laplacian naturally arises in a class of important problems, in particular in random walks on networks, as $D^{-1}A$ is the transition matrix for probability distributions arising from such walks [3]. The eigenvectors of L form a complete set that span \mathbb{R}^n , and the eigenvalues λ_i are real numbers and can be ordered as

$$0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n \leq 2.$$

In particular, the upper and lower bounds imply that

$$|1 - \lambda_k| \leq 1, \quad k = 1, 2, \dots, n. \quad (10)$$

The first eigenvalue λ_1 is always zero and corresponds to the eigenvector $(1, 1, \dots, 1)^\top$. The second eigenvalue λ_2 , which is also called the *spectral gap*, is positive if and only if the graph is connected. In fact, the multiplicity the zero eigenvalue equals the number of connected components of the graph. In a connected graph, λ_2 gives an indication of how difficult it is to disconnect the graph into two large pieces by removing a small number of edges, and is thus directly related to graph connectivity.

In matrix form, (8) becomes

$$\dot{x}(t) = D^{-1}A(2x(t) - x(t-\tau)) - x(t), \quad (11)$$

with $x = (x_1, x_2, \dots, x_n)^\top$. Since L has a complete set of eigenvectors $\{\mathbf{v}_k\}$, one can write $x(t) = \sum_{k=1}^n u_k(t)\mathbf{v}_k$ for

some scalar coefficients u_k , which transforms (11) into a system of n decoupled scalar equations

$$\dot{u}_k(t) = (1 - 2\lambda_k)u_k(t) - (1 - \lambda_k)u_k(t - \tau), \quad k = 1, \dots, n. \quad (12)$$

The characteristic equation corresponding to the eigenmode (12) is

$$\psi_k(s) := s - 2(1 - \lambda_k) + 1 + (1 - \lambda_k)e^{-s\tau} = 0. \quad (13)$$

Consequently, the characteristic equation for the whole system (11) can be written as

$$\Psi(s) := \prod_{k=1}^n \psi_k(s) = 0. \quad (14)$$

Note that $s = 0$ is always a characteristic root for the first factor $\psi_1(s) = s - 1 + e^{-s\tau}$ corresponding to the first eigenmode, $\lambda_1 = 0$. Thus, points on the synchronization subspace spanned by $\mathbf{v}_1 = (1, 1, \dots, 1)^\top$ can be at best neutrally stable. If that is the case, and in addition $\lim_{t \rightarrow \infty} u_k(t) = 0$ for all $k \geq 2$, then the system converges to a point on \mathbf{v}_1 , i.e. it reaches consensus, from arbitrary initial conditions. This clearly happens if and only if zero is a simple root of Ψ and all other roots of Ψ have negative real parts. Moreover, in this case the speed of convergence from general initial conditions depends on the slowest of these modes u_k , $k \geq 2$. Hence, we factor (14) into directions along and transverse to the synchronization subspace $\text{span}(\mathbf{v}_1)$ as $\Psi(s) = \psi_1(s)\tilde{\Psi}(s)$, where

$$\tilde{\Psi}(s) := \prod_{k=2}^n \psi_k(s), \quad (15)$$

and use the transverse part to quantify the speed of convergence, which motivates the following definition.

Definition 1: The number $s^* \in \mathbb{C}$ is called *the dominant transverse root of the consensus algorithm* (or *dominant root*, for short) if $\tilde{\Psi}(s^*) = 0$ and all roots s of $\tilde{\Psi}$ satisfy $\text{Re}(s) \leq \text{Re}(s^*)$.

When $\tau = 0$, system (8) reduces to the classical consensus problem (1)–(2), and the characteristic equation (14) reduces to

$$\Psi(s) = \prod_{k=1}^n (s + \lambda_k); \quad (16)$$

so, the dominant root is given by the second smallest eigenvalue λ_2 of the Laplacian. Therefore, λ_2 and the algebraic connectivity of the graph play an important role for convergence speed (how fast the network reaches consensus), small values of λ_2 implying a slow convergence. Note that λ_2 can be arbitrarily small in connected networks, depending on the connection structure.

III. CONVERGENCE OF THE CONSENSUS ALGORITHM

We begin by some observations on the roots of a certain complex function.

Lemma 2: The function

$$\psi(s) = s - \beta(1 - e^{-s}), \quad \beta \in \mathbb{R}, \quad (17)$$

has a simple root at zero and all its other roots have negative real parts if and only if

$$\beta < 1. \quad (18)$$

Proof: Clearly $\psi(0) = 0$; so zero is always a root of ψ . Moreover, $\psi'(0) = 1 - \beta$ is nonzero if and only if $\beta \neq 1$, in which case zero is a simple root. We note that $\psi(s)$ is a special case of the function

$$\tilde{\psi}(s) := s - a_1 - a_2 e^{-s}, \quad a_1, a_2 \in \mathbb{R}. \quad (19)$$

which has been studied in the classical paper of Hayes [8]. The properties for (19) is therefore well-known; here we recall the stability region depicted in Figure 2. In particular, for the parameter values on the semi-infinite line $L = \{a_1, a_2 : -a_2 = a_1 < 1\}$ in the figure, $\tilde{\psi}$ has a simple root at zero and all its remaining roots have negative real parts. Since ψ is a special case of (19) with $a_1 = -a_2 = \beta$, condition (18) follows from considering the line L in Figure 2. ■

We can now state the main convergence result for the consensus problem (8) of anticipating agents.

Theorem 3: The system (8) defined on a connected undirected graph reaches consensus from arbitrary initial conditions if and only if

$$\tau < 1. \quad (20)$$

Proof: We will show that the characteristic equation (14) has a simple root at zero and all the remaining roots have negative real parts if and only if condition (20) holds. Consider first the roots of the first factor $\psi_1(s) = s - 1 + e^{-s\tau}$ in (14). By a change of variable $s' = s\tau$, we can equivalently consider the roots of

$$\hat{\psi}_1(s') = s' - \tau + \tau e^{-s'}. \quad (21)$$

Lemma 2 gives that $\hat{\psi}_1$, and therefore ψ_1 , has a simple root at zero and all the other roots have negative real parts if and only if (20) holds. It then suffices to show that all roots of the remaining factors ψ_k , $k = 2, \dots, n$, in (14) have negative real parts under condition (20). Now, the roots s of ψ_k satisfy

$$s = a_k + b_k e^{-s\tau}, \quad (21)$$

with

$$a_k = 2(1 - \lambda_k) - 1 \quad (22)$$

$$b_k = -(1 - \lambda_k). \quad (23)$$

For $\tau = 0$, (21) reduces to

$$s = a_k + b_k = -\lambda_k < 0 \quad \text{for } k \geq 2, \quad (24)$$

and the roots are on the open left complex plane. We next look for roots that may cross the imaginary axis as τ is increased from zero. Letting $s = i\omega$, $\omega \in \mathbb{R}$, the imaginary part of (21) gives

$$\omega = -b_k \sin(\omega\tau).$$

Therefore,

$$|\omega| = |b_k \sin(\omega\tau)| \leq |b_k \omega\tau|. \quad (25)$$

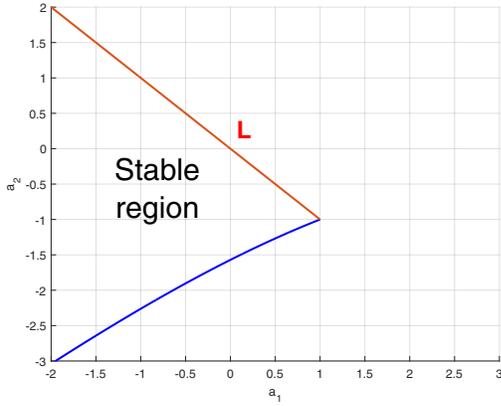


Fig. 2. Stability region of equation (19) in a_1 - a_2 parameter space.

Notice from (21) and (24) that $\omega \neq 0$. Hence, dividing (25) by $|\omega|$,

$$1 \leq |b_k \tau| = |\tau(1 - \lambda_k)| \leq \tau,$$

where we have substituted for b from (23) and used (10). This shows that, as long as condition (20) holds, no roots can cross the imaginary axis, and so all roots of ψ_k for $k = 2, \dots, n$ have negative real parts. This completes the proof of the theorem. ■

IV. COMPUTATION OF DOMINANT ROOTS

The performance of the consensus algorithm depends on the transverse dominant root of the problem, as defined in Definition 1. We use Lambert W function to solve for the characteristic roots.

Recall that the Lambert W function is defined as the inverse function of the mapping $f(z) = ze^z$ for $z \in \mathbb{C}$ [4]; in other words, $W(z)$ satisfies

$$W(z)e^{W(z)} = z. \quad (26)$$

Since f is not one-to-one, $W(z)$ is multi-valued. We let W_0 denote the principal branch of the Lambert function.

Proposition 4: The root of the characteristic equation (13) having the largest real part is given by

$$s = \frac{1}{\tau} W_0(\tau b_k e^{-a_k \tau}) + a_k, \quad (27)$$

where W_0 is the principal branch of the Lambert W function and a_k and b_k are defined in (22)–(23). Consequently, the dominant transverse root of the consensus problem equals

$$\max_{2 \leq k \leq n} \left\{ \frac{1}{\tau} W_0(\tau b_k e^{-a_k \tau}) + a_k \right\}. \quad (28)$$

Proof: As seen in the proof of Theorem 3, the roots of ψ_k satisfy equation (21), which can be re-written as

$$s - a_k = b_k e^{-(s-a_k)\tau} e^{-a_k \tau}.$$

A change of variables $s' \rightarrow (s - a_k)\tau$ gives

$$s' = \tau b_k e^{-a_k \tau} e^{-s'},$$

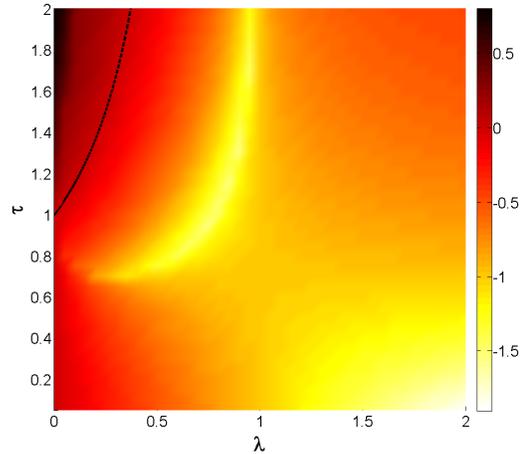


Fig. 3. Map of the dominant characteristic root of eigenmodes corresponding to a generic Laplacian eigenvalue λ and delay value τ . The colour represents the real part of the dominant root of a factor (13) of the characteristic equation. The region above the black line is when there is a root with positive real part. Note that, since zero is always an eigenvalue of the Laplacian in every network, the condition $\tau < 1$ actually defines the region for reaching consensus, in accordance with Theorem 3.

from which, by the definition (26) of Lambert function, one immediately has $s' = W(\tau b_k e^{-a_k \tau})$. Going back to the original variable s shows that the roots of ψ_k are given by

$$s = \frac{1}{\tau} W(\tau b_k e^{-a_k \tau}) + a_k,$$

It follows by a recent result from [15] that the root with largest real part is given by the principal branch of the Lambert function, which establishes (27). Definition 1 then implies (28). ■

Using Proposition 4 and equation (27), we calculate the dominant root for any eigenmode (12) of system (8) corresponding to a generic Laplacian eigenvalue λ . This gives a universal map of the dominant roots of the consensus problem (8) for any graph topology and delay value, which is depicted in Figure 3.

In the color map of Figure 3, lighter colors correspond to more negative real parts for the roots. Hence one notes, from the change in color as one moves vertically upwards from the horizontal axis, that a positive value of τ can indeed improve convergence speed for a given eigenmode. In the next section we will confirm this observation through actual simulation of several networks.

V. SIMULATION RESULTS

We illustrate and elaborate on the analytical results using numerical simulations in three example networks with various connection topologies. We compare the performance of the delayed anticipatory algorithm with the classical consensus algorithm. In all pairwise comparisons simulations are started from random initial conditions but identical ones for both algorithms.

Network 1: Circular network made up of 20 vertices, where each vertex has two neighbours, one on each side (see

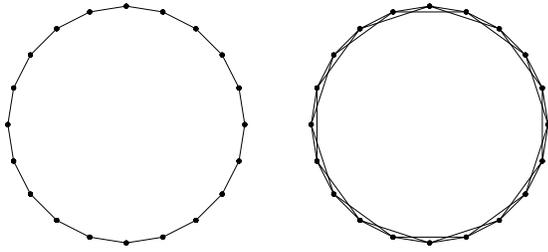


Fig. 4. Network 1 (left) and Network 2 (right) used in the simulations. Each one is a regular network of 20 vertices arranged on a circle, where each vertex has 2 or 4 neighbors, respectively.

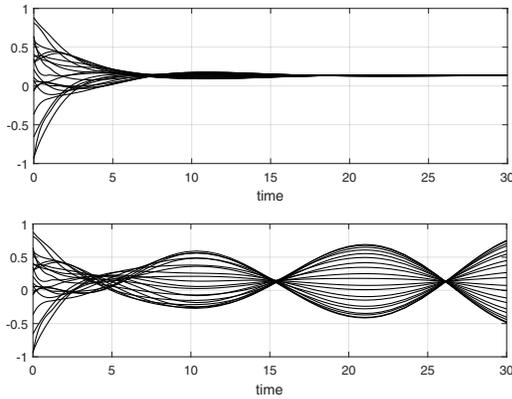


Fig. 5. Time evolution of agents' states in (8) in the configuration of Network 1 starting from arbitrary initial conditions. In the top figure $\tau = 0.9$ and the system converges towards consensus, whereas in the bottom figure $\tau = 1.1$ and the system diverges.

Figure 4).

We first check the converge condition (20) of Theorem 3. Figure 5 shows the comparison between results obtained with $\tau = 0.7$ and $\tau = 2$, respectively, for which the system converges or diverges, affirming condition (20). The convergence criterion depends only on τ and is independent of the network structure, as long as the latter is connected.

We next take $\tau = 0.7$ for Network 1 and compare it with the classical undelayed algorithm $\tau = 0$. Figure 6 shows that the speed of convergence to consensus is much faster under the anticipatory algorithm. This is confirmed from another point of view in Figure 7, which plots the standard deviation of the agents' states over time. The comparison makes it clear that the anticipatory algorithm performs much better. We note that the circular graph configuration has a small spectral gap λ_2 , and therefore converges poorly under the classical undelayed consensus protocol. However, the same network exhibits a dramatic improvement in convergence under the anticipatory protocol with $\tau = 0.7$.

Network 2: Circular network made up of 20 vertices, where each vertex has four neighbours, two on each side (see Figure 4). This network is similar to the initial setup of used in [11], where the author demonstrated that convergence to consensus can be improved by random rewirings of the

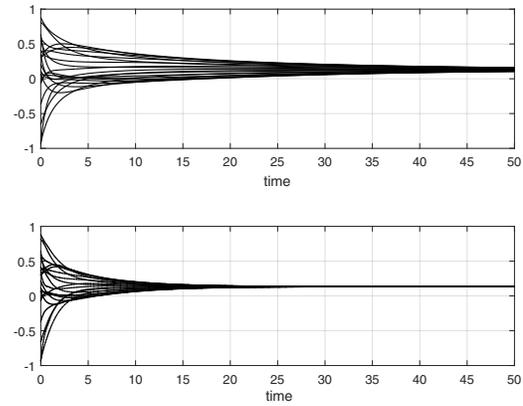


Fig. 6. Time evolution of agents' states in Network 1 under the classical consensus protocol obtained with $\tau = 0$ (top figure) and the anticipatory protocol obtained with $\tau = 0.7$ (bottom figure), showing that the system reaches consensus much faster under the anticipatory algorithm.

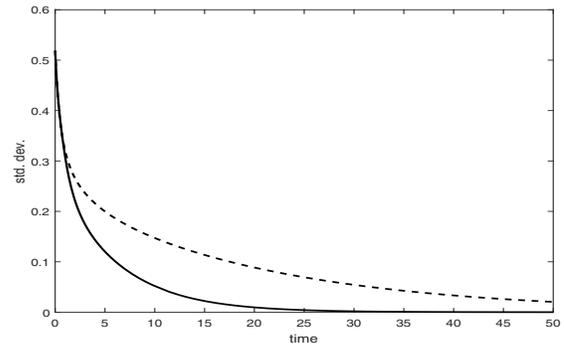


Fig. 7. Time evolution of the standard deviation of agents' states in Network 1, plotted for $\tau = 0$ (dashed line) and $\tau = 0.7$ (solid line), showing the superior performance of the anticipatory algorithm.

links en route to a small-world network configuration. Here, we will accomplish a similar improvement by using the anticipatory consensus algorithm instead of rewiring the network. Figure 8 shows the time evolution of the agents' states in Network 3 under the anticipatory algorithm with $\tau = 0.5$, and compares it with the classical consensus algorithm with $\tau = 0$. It is seen that the convergence speed is increased without the need to rewire the network.

Network 3: An Erdős-Rényi random undirected graph with 20 nodes. If the edges of Network 2 are randomly rewired, the consensus performance increases monotonically at each step of rewiring, as shown in [11]. After repeated random rewirings, the network approaches a random network, which should then have very good convergence properties. We therefore generate a random network with (approximately) the same number of edges as in Network 2, which is accomplished by taking $p = 0.21$ as the probability of having an edge between a pair of vertices. Figures 9 and 10 show, once again, that consensus is reached faster under the anticipatory algorithm. We conclude that using an anticipatory algorithm

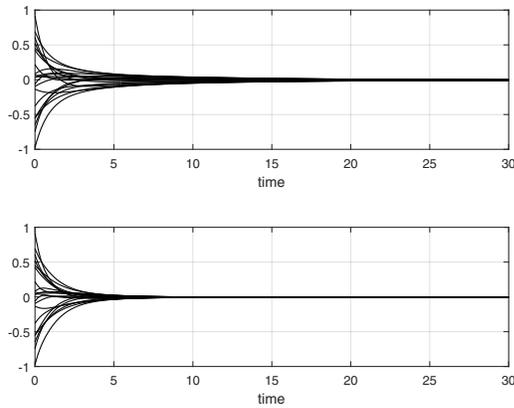


Fig. 8. Time evolution of agents' states in Network 2 under the classical consensus protocol (top figure) and the anticipatory protocol with $\tau = 0.5$ (bottom figure), demonstrating the superior performance of the latter algorithm.

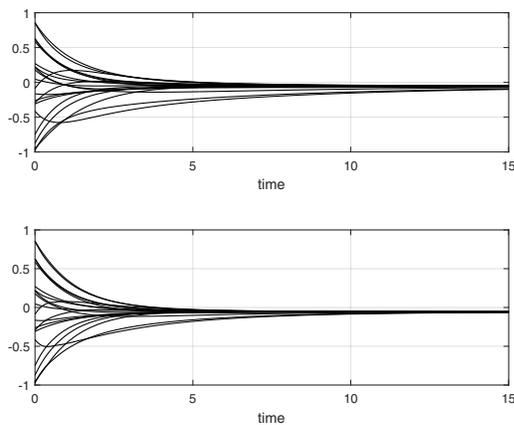


Fig. 9. Time evolution of agents' states in Network 3 under the classical consensus protocol (top figure) and the anticipatory protocol with $\tau = 0.5$ (bottom figure).

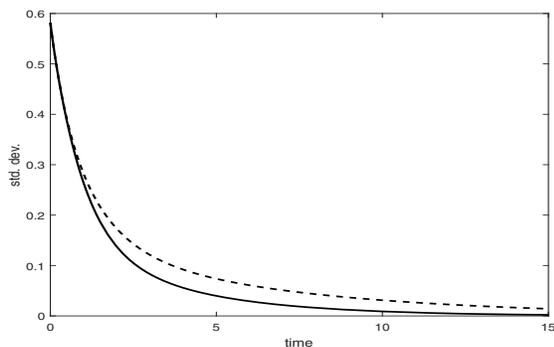


Fig. 10. Standard deviation of agents' states in Network 3 obtained under the classical consensus protocol (dashed line) and under the anticipatory protocol with $\tau = 0.5$ (solid line).

can lead to a better consensus performance than rewiring the network, which is useful because altering the network structure is not always possible in real networks. For reasons of clarity we have illustrated the results on networks with 20 agents, but the results are similar for larger networks.

ACKNOWLEDGMENT

The authors thank the ZiF (Center for Interdisciplinary Research) of Bielefeld University for their support within the cooperation program *Discrete and Continuous Models in the Theory of Networks*, where this problem was initiated. Part of the research was carried out when DI visited the Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany, in Summer 2015 under financial support from DAAD Short-Term Grants 2015 (57130097) and from L2S Laboratory, France. FMA acknowledges support from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 318723 (MATHEMACS).

REFERENCES

- [1] F. M. Atay. Consensus in networks under transmission delays and the normalized Laplacian. In R. Sipahi et al., editor, *Time Delay Systems – Methods, Applications and New Trends*, volume 423 of *Lecture Notes in Control and Information Sciences*, pages 407–416. Springer-Verlag, Berlin, 2012.
- [2] F. M. Atay. The consensus problem in networks with transmission delays. *Philos. Trans. R. Soc. A*, 371:20120460, 2013.
- [3] Fan R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 1997.
- [4] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.
- [5] M. H. DeGroot. Reaching a consensus. *J. Amer. Statist. Assoc.*, 69(345):118–121, 1974.
- [6] J. A. Fax and R. M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control*, 49(9):1465–1476, 2004.
- [7] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer-Verlag, New York, 2001.
- [8] N. D. Hayes. Roots of the transcendental equation associated with a certain difference-differential equation. *J. London Math. Soc.*, 25:226–232, 1950.
- [9] N. A. Lynch. *Distributed Algorithms*. CA: Morgan Kaufmann, San Francisco, 1996.
- [10] L. Moreau. Stability of continuous-time distributed consensus algorithms. In *Proc. 43rd IEEE Conference on Decision and Control*, pages 3998–4003, 2004.
- [11] R. Olfati-Saber. Ultrafast consensus in small-world networks. In *ACC: Proceedings of the 2005 American Control Conference, Vols 1-7*, pages 2371–2378, 2005.
- [12] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Proc. 2005 European Control Conference*, volume 49, pages 6698–6703, 2005.
- [13] Reza Olfati-Saber and Richard M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automat. Control*, 49(9):1520–1533, 2004.
- [14] A. Seuret, D. V. Dimarogonas, and K. H. Johansson. Consensus under communication delays. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4922–4927. IEEE, 2008.
- [15] H. Shinozaki and T. Mori. Robust stability analysis of linear time-delay systems by Lambert W function: Some extreme point results. *Automatica*, 42:1791–1799, 2006.
- [16] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75(6):1226–1229, 1995.