Neurocomputing 570 (2024) 127130

Contents lists available at ScienceDirect

Neurocomputing

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

Convergence speed of dynamic consensus with delay compensation

Rosario Aragues [a,*], Antonio González [b], Gonzalo López–Nicolás [a], Carlos Sagues [a]

[a] Universidad de Zaragoza and Instituto Universitario de Investigación en Ingeniería de Aragón I3A, Spain
[b] Instituto de Automática e Informática Industrial (AI2), Universitat Politècnica de València, Valencia, Spain

ARTICLE INFO

Communicated by D. Ding

Keywords:
Sensor networks
Dynamic average consensus
Time-delays
Distributed and cooperative systems
Distributed control of multi-agent systems

ABSTRACT

A well-known drawback in distributed average consensus of multi-agent systems is that the exchanged information is usually delayed due to the time elapsed during the data transmission process. Using classical dynamic average consensus, delays may lead to poor performance or even instability. In this paper, we propose a novel dynamic consensus method that counteracts the negative effects of delays by means of delay compensation techniques. The interest of our dynamic consensus method with delay compensation is that it converges under mild conditions on graph connectivity and bounded reference signals, no matter how large the delays are, as long as delays are fixed and known. We also provide a formal characterization of the convergence speed of our method. Additionally, our results apply to fixed directed strongly connected, and undirected topologies.

1. Introduction

The problem of distributed average consensus [1–4] is an interesting topic of research with a variety of applications, including formation control [5], convex optimization [6], economic dispatch [7], or target tracking [8], among others. In dynamic average consensus scenarios, a team of robots measures a set of reference signals (possibly time-varying) and can exchange data with their neighbors (nearby robots). In this case, each robot updates its state based on the exchanged information with the objective of cooperatively tracking the averaged consensus of the reference signals. The problem of static consensus is similar, but considering that the reference signals are time-invariant.

One relevant problem to consider in this framework is the presence of delays, which may cause degradation or even instability if they are not considered in the design process. Some contributions in the context of neural network systems can be found in [9–12]. Apart from time delays, different phenomena have been included in the stability analysis, such as time-varying uncertainties in the system model, disturbances in the measured outputs [13], or nonuniform trial lengths [14]. In this respect, further contributions have investigated robust control design methodologies aimed at reducing the negative impact of time delays, disturbance signals, and model uncertainties [13–15].

A multi-robot team constitutes a networked system, in which the agents interact via communications. A known drawback in networked systems is that the exchanged information is usually delayed due to the time elapsed during the data transmission process [16–19]. These delays may jeopardize the closed-loop system performance, and may even lead to instability if they are large enough. This motivated the convergence analysis of consensus algorithms in the presence of time delays. It is well known that there exists a maximum allowable delay under which the system remains stable for consensus problems, such as group consensus [20], static consensus [21–29], and dynamic consensus [30]. In the particular case that delays only affect the received measurements coming from neighbor agents or robots, then stabilization is guaranteed irrespective of the size of delays [21,31–35], although these delays can lead to slower convergence or poor performance [31].

Instead, it would be desirable to count on strategies aimed at reducing the negative effects of delays and ensuring convergence under the same conditions on the topological structure as the ones associated with the delay-free static or dynamic consensus strategies. In this paper, we study a dynamic consensus algorithm under delayed data. Opposite to [21,31–35], delayed *relative* measurements between neighbors are here considered. This implies that there exist delays in the diagonal term of the overall closed-loop system matrix (i.e., we are not restricted to the case that delays only affect the off-diagonal terms of the overall closed-loop system matrix). Thus, we need to provide additional mechanisms to mitigate the negative effects of the delays. In this respect, some contributions proposed to let each robot predict the current states of the neighbors and use this predicted data to carry out its updates [36]. Nevertheless, these approaches have the inconvenience of high memory consumption and computation complexity. Since one of the strengths of using distributed strategies is their inherent scalability, it would be desirable to decrease to the greatest extent the required resources in

* Corresponding author.
E-mail addresses: raragues@unizar.es (R. Aragues), angonsor@upv.es (A. González), gonlopez@unizar.es (G. López–Nicolás), csagues@unizar.es (C. Sagues).

https://doi.org/10.1016/j.neucom.2023.127130
Received 22 May 2023; Received in revised form 16 September 2023; Accepted 6 December 2023
Available online 13 December 2023

terms of memory usage and computational cost. Another interesting method is to face communication delays by using predictor-feedback delay compensation approaches [37–39]. The underlying idea behind delay compensation is to use current and past information of the system process to predict the future state with a prediction horizon equivalent to the delay value (but without explicitly keeping predictions of the states of all neighbors, as in [36]). Thus, the state feedback control can make use of the prediction of the state variable to counteract the effect of delays in the closed-loop system. Delay compensation requires prior knowledge of the system model in order to find an equivalent delay-free representation, with the advantage that control synthesis in the presence of time delays can be simplified, that is to say, the controller parameters can be designed by traditional methods without considering delays. In distributed scenarios like the one considered here, it is generally not possible to obtain an exact prediction of the system. However, as discussed in this paper, partial delay compensation can be achieved by introducing extra parameters in our predictor-feedback delay compensation strategy, preventing the system from becoming unstable. Differently from conventional predictor-feedback methods, each robot only uses its current and past local inputs to obtain a prediction of the system state. Since only local information is available to each agent, such prediction cannot be exact but can lead to some improvement if the prediction is properly weighted.

In the context of distributed robot networks, few works [5,40–42] resort to predictor-feedback methods for delay compensation: In [40], they consider static consensus problems and include parameters to weight the importance of terms involving historical input data, giving also the limits on these parameters to ensure convergence for arbitrarily large delays. In [41], delays are compensated in a static consensus law of high-order systems by means of truncated predictor feedback. The effectiveness is proved for any arbitrarily large delay provided that they are time-constant and known. In [43], the Artstein's reduction method is adapted to obtain a delay-free model in static consensus scenarios for continuous time systems. The work [42] proposes a predictor feedback scheme by considering agents with linear continuous-time dynamics models. The consensus problem addressed there studies the convergence of the local state variables of each agent to a common value, which is not necessarily the average, but does not discuss the average tracking control problem, as we propose here.

In our previous work [5], we successfully used the historical information of the control inputs to increase the robustness of formation control methods under time-varying delays and fixed graphs. The stability analysis was addressed by means of LMI conditions obtained by the Lyapunov method. Compared to our previous work [5], here we consider dynamic consensus scenarios. We carefully study the relation between the convergence speed and eigenvalues of the underlying graph, and the overall closed-loop system matrix in the presence of delays.

The solutions proposed in [5,40–42] do not fully solve the problem considered in this paper. First, these works do not deal with prediction feedback and delay compensation applied to the problem of *dynamic average consensus*, where the aim is not only to achieve consensus but also to ensure the consensus value *tracks* the time-varying average of the references, as explained in the paper. As we discuss later in Section 6.3, dealing with time-varying reference signals, presents challenges that are inherent to the dynamic average problem. Second, in this work we are interested in obtaining closed-form expressions on the parameter configuration and on the convergence speed (transient behavior). We also provide insight into how delays affect the properties of the overall closed-loop system matrix and their effects on both transient and steady-state performance. Indeed, not only mere stability but also dynamic performance is relevant in practice. For instance, [30,31] focused on analyzing the transient response in respectively static and dynamic consensus, but without considering any delay compensation strategy.

The main contributions of this paper are:

- We propose a dynamic consensus method for discrete-time systems with a predictor feedback strategy for delay compensation. As far as we know, this approach has not been previously applied to any dynamic consensus algorithm.
- The proposed method can cope with a high variety of graphs, including undirected graphs that may be sparsely connected, as well as directed graphs that are strongly connected.
- We ensure convergence for arbitrarily large time delays expressed in time steps (d), provided that the equivalent delay-free closed-loop system is stable.
- We provide analytic expressions that reveal the influence of communication topology and delays on the speed of convergence. These results rely on a thorough eigenvalue-based analysis of the performance of the dynamic consensus method.
- We propose parameter tuning rules aimed at improving the speed of convergence. Remarkably, this adjustment does not depend on the delay value.
- The method is scalable, and it has light requirements in terms of communication, computational, and memory costs.

This paper is organized as follows. Section 2 states the problem and introduces the notation used in the paper. Section 3 presents the proposed dynamic consensus algorithm with delay compensation. Section 4 presents the proposed algorithm under different alternative representations that are useful later for convergence analysis. Section 5 studies the properties of the method in terms of its convergence speed. In this section, we state the main result of the paper, regarding the stability and transient behavior of the proposed method. Section 6 presents several simulations with different networks and delay values, and Section 7 gives the conclusions of the paper.

2. Preliminaries and problem description

Let $\mathbf{I}_n$ be the $n \times n$ identity matrix, $\mathbf{0}_{n_1 \times n_2}$ be an $n_1 \times n_2$ matrix with all entries equal to 0, and $\mathbf{1}_n$ and $\mathbf{0}_n$ be column vectors with all entries equal to 1 and 0 respectively. The dimensions are omitted when they can straightforwardly be inferred.

There are n robots in an environment. At every step $k \in \mathbb{Z}$ robot i has a state $x_i(k) \in \mathbb{R}$ and it can modify its state according to the control input $u_i(k)$, discussed later, and the update rule $x_i(k) = x_i(k-1) + u_i(k)$, for $i = 1, \dots, n$. Robots exchange data according to a *directed* communication graph (*digraph*) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, n\}$ are the robots, and there is an edge $(i, j) \in \mathcal{E}$ if robot i can receive data from robot j, i.e., if robot j is one of the *neighbors* or robot i. We assume that the directed graph $\mathcal{G}$ is *strongly connected*: there exists a directed path between every pair of robots (all robots can receive data from all the other robots in a multi-hop way). Let matrices $\mathcal{A} \in \{0,1\}^{n \times n}$ and $\mathcal{L} \in \mathbb{R}^{n \times n}$ be respectively the adjacency matrix and the Laplacian matrix associated with $\mathcal{G}$,

$$\mathcal{A} = [a_{ij}] = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E}, i \neq j, \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathcal{L} = \text{diag}(\mathcal{A}\,\mathbf{1}) - \mathcal{A}, \tag{1}$$

and we let $\mathcal{N}_{\max} = \max_{i \in \{1,\dots,n\}} \{\sum_{j \neq i}$ be the maximum degree.

Each robot i, for $i \in \{1, \dots, n\}$, measures a reference signal $r_i(k)$, that varies over time k. The *dynamic average consensus* problem consists of designing control inputs $u_i(k)$ that only depend on local data (current and previous robot states $x_i(k')$ and references $r_i(k')$, $k' \leq k$), and on information from the neighbors (agents j with $(i, j) \in \mathcal{E}$, or equivalently, $a_{ij} = 1$) so that every robot tracks the weighted average (for strongly connected digraphs) of the references,

$$r_{wavg}(k) = \frac{w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \mathbf{r}(k), \tag{2}$$

where $\mathbf{r}(k) = \left[r_1(k), \dots, r_n(k)\right]^T$, and $w_{\mathcal{L}_1}$ is the left eigenvector of the Laplacian matrix (1) associated with the eigenvalue 0, i.e., $w_{\mathcal{L}_1}^T \mathcal{L} =$

$\mathbf{0}^T$. In undirected graphs and weight-balanced digraphs, the adjacency matrix in (1) has equal row and column sums [25, Definition 1], and they have the property that $\mathbf{1}^T$ is a left eigenvector of the Laplacian $\mathcal{L}$ associated with the eigenvalue 0, i.e., $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$. Thus, (2) is the exact average,

$$r_{avg}(k) = \frac{1}{n}\sum_{j=1}^{n} r_j(k). \tag{3}$$

In strongly connected digraphs, $w_{\mathcal{L}1}^T$ is not necessarily equal to $\mathbf{1}^T$, and this vector introduces a weighting factor in the consensus value in (2) [25].

An example of a dynamic average consensus algorithm follows [1,2,30]:

$$\begin{aligned} x_i(k) &= x_i(k-1) + u_i(k), \text{ for } k \geq 1 \\ u_i(k) &= r_i(k) - r_i(k-1) - \beta\delta \sum_{j=1}^{n} a_{ij}(x_i(k-1) - x_j(k-1)), \\ x_i(0) &= r_i(0), \end{aligned} \tag{4}$$

where β and δ are controller parameters that must be tuned. The static average consensus [25] is a particular case of (4), where the reference signals remain constant along the iterations ($r_i(k) \equiv r_i$ for all k), and the robot states converge to the average of the initial states: $(1/n)\sum_{j=1}^{n} x_j(0)$. Both in static and dynamic consensus methods (4), the control input $u_i(k)$ is computed using information locally available to agent i and information received from the neighbors (note from (4) that a_{ij} in (1) equals 0 for non-neighbor agents, i.e., $a_{ij} = 1$ iff $(i,j) \in \mathcal{E}$). In dynamic consensus methods, each agent i computes the current state $x_i(k)$ by means of (4) in two steps: first, the current value of the time-varying reference signal $r_i(k)$ is measured and second, the control input $u_i(k)$ is computed using the measured value of $r_i(k)$ together with the information received from the neighbors. Note that agent i also needs to recall the previous state and reference signal at step $k-1$ $(x_i(k-1), r_i(k-1))$.

In the absence of delays [1,2], parameters β and δ must satisfy the following conditions in order to ascertain the convergence of the consensus law (4):

$$\delta \in \left(0, \frac{1}{\beta\mathcal{N}_{\max}}\right), \qquad \beta > 0. \tag{5}$$

Then, for connected undirected graphs, and references that experience bounded variations, the states of every agent $x_i(k)$, for $i = 1, \ldots, n$ converge asymptotically to a neighborhood of the average of the reference signals, satisfying [1,2]:

$$\begin{aligned} &\limsup_{k\to\infty} \left| x_i(k) - \frac{1}{n}\sum_{j=1}^{n} r_j(k) \right| \leq \frac{\alpha}{\beta\delta\lambda_{\mathcal{L}_2}}, \\ &\alpha = \max_k \left(\left\| (\mathbf{I} - \mathbf{1}\mathbf{1}^T/n)(\mathbf{r}(k) - \mathbf{r}(k-1)) \right\|_2 \right), \end{aligned} \tag{6}$$

$\lambda_{\mathcal{L}_2}$ is the second smallest eigenvalue of the Laplacian, and $\mathbf{r}(k) = \left[r_1(k), \ldots, r_n(k)\right]^T$ as in (2). Similar results can be found in [1,2,30] for strongly connected weight-balanced directed graphs (where robots track the average), or for strongly connected digraphs (where robots track a *weighted* average [25]).

2.1. Consensus in the presence of delays

We consider a scenario where the reference $r_i(k)$ is immediately perceived by robot i, but the received data from neighbor agents are affected by communication delays.

Assumption 1 (*Delays*). For our analysis, we consider fixed and known delays expressed in time steps d which are common to all communication links between agents, i.e., agents receive the data sent by their neighbors d steps later.

Assumption 2 (*Communication Graph*). We assume that the communication graph $\mathcal{G}$ is directed, fixed (time-constant), and strongly connected. Note that this includes undirected connected graphs as a special case.

Assumption 3 (*Bounded Reference Variations*). We assume that the reference signals $\mathbf{r}(k)$ experience bounded variations over time, i.e., that there exists α_∞^+ finite such that it bounds the variations of references along different steps.

Note that Assumption 1 appears often in consensus problems affected by time-delays [16–19,43]. Assumption 3 is also similar to the ones that appear in classical dynamic consensus methods. The particular expression for α_∞^+ in our case will be clearer when it is used later in Section 5 (Proposition 3, Theorem 1 (64)), once the algorithm has been presented. The main aspect of this Assumption is to ensure that there exists *some* value that bounds the variations of the reference signals. Assumption 2 imposes conditions that are mild on the connectivity of the graph. This conditions allows a wide variety of connectivity graphs, including undirected connected graphs that may be sparse. It also includes digraphs strongly connected.

In the presence of delays, the immediate application of the previous control laws would lead to:

$$\begin{aligned} u_i(k) &= r_i(k) - r_i(k-1) \\ &\quad - \beta\delta \sum_{j=1}^{n} a_{ij}\left(x_i(k-1-d) - x_j(k-1-d)\right), \end{aligned} \tag{7}$$

for the dynamic consensus case in (4). Recall that $r_i(k)$ is the current measurement of the reference signal, obtained when agent i is going to compute the current state $x_i(k)$, whereas $r_i(k-1)$ is the previous reference signal, measured when the previous state $x_i(k-1)$ was computed. It is well known that the static consensus method under delays d as in (7) (with $r_i(k) = r_i(k-1)$) converges only if the delays are sufficiently small [21–29]. For instance, in the case of discrete-time systems with undirected graphs and $\beta\delta = 1$, [24] gives the following maximum allowable delay bound for d in (7):

$$d < \frac{1}{2}\left(\frac{\pi}{2\arcsin(\mathcal{N}_{\max})} - 1\right), \tag{8}$$

where $\mathcal{N}_{\max}$ is the maximum degree of the graph. On the other hand, if the delay d in (7), instead of affecting the relative data $\left(x_i(k-d) - x_j(k-d)\right)$, only affects the global information $x_j(k-d)$ received from neighbors, i.e., if the control input depends on $\left(x_i(k) - x_j(k-d)\right)$ in (7), then static consensus methods can cope with arbitrarily large delays [21,31–35].

More recently, an analytic expression for the maximum allowable input delay for stability in dynamic consensus (7) was provided in [30, Lemma III.4] for undirected graphs:

$$d < \min_{i\in\{2,\ldots,n\}} \frac{1}{2}\left(\frac{\pi}{2\arcsin(\frac{\beta\delta\lambda_{\mathcal{L}_i}}{2})} - 1\right), \tag{9}$$

where $\lambda_{\mathcal{L}_i}$ for $i \in \{2, \ldots, n\}$ are the nonzero eigenvalues of the Laplacian $\mathcal{L}$. Similar expressions appear in [30] for strongly connected weight-balanced digraphs. The delay also affects the tracking error [30, Theorem III.2] in (6), which now depends not only on α, β, δ but also on additional parameters that can be obtained using Linear Matrix Inequalities on the delayed system. The reader is referred to [30] for a full explanation.

Problem formulation 1. Our goal is to design the control inputs $u_i(k)$ for the dynamic consensus method (4), under delays in the communications (Assumption 1) that affect the relative states of the agents, $(x_i(k-1-d) - x_j(k-1-d))$, so that the convergence is ensured for all possible values of the delay d under the same conditions imposed by the underlying communication graph $\mathcal{G}$. We consider strongly connected digraphs, which include undirected graphs as a particular case.

Note that this could be done in a straightforward way if we extended to the dynamic case the static consensus ideas [21,31–35] of imposing that the local agent's state is not subject to time delays (i.e. if the relative measurements in (7) are $(x_i(k-1)-x_j(k-1-d))$). Here, however, we let the relative information $(x_i(k-1-d)-x_j(k-1-d))$ be delayed. In order to counteract the negative effects of delays in the closed-loop performance of the dynamic consensus methods, we propose to use a scheme based on delay compensation by introducing past information [5,39,40].

The historical information of the control inputs was successfully used in [40] for the static consensus case. In our previous work [5], we used this historical data as well to improve the closed-loop dynamic performance in the presence of delays in formation control synthesis. However, as far as we know, these ideas have not been previously applied to the dynamic consensus problem. Moreover, in this paper, we address a novel study of the transient behavior and we provide an accurate characterization of the relation between the eigenvalues of the Laplacian matrix of the underlying topology, and the convergence speed of the delayed system. In addition, although we are interested in undirected graphs (tracking of the average), in order to make the proposal more general, we provide results for strongly connected digraphs (tracking of a weighted average), which include undirected graphs as a particular case.

2.2. Examples of application

Fig. 1 shows two examples of applications of the problem addressed in which our proposed method would be of interest. The image at the top shows a scenario with a wireless sensor network deployed in an area for monitoring a phenomenon of interest (illustrated with background colors). Circular nodes are equipped with sensors to measure the reference signals. Squares represent *relay nodes* [44,45], that do not carry sensors but are just in charge of keeping a multi-hop communication between sensor nodes. Nodes periodically wake up and perform one round of operations (take one measurement/transmit a 1-hop message) and sleep again, to save batteries and increase the lifetime of the sensor network. Here, d represents the number of hops between the sensing nodes. There are d relay nodes between each pair of neighboring sensor nodes, by construction.

In Fig. 1, the images at the bottom represent another scenario in which node i takes a relative measurement of node j. The required time for processing the relative measurement is dT, whereas it is desired to sample the reference signals with period T. Thus, each node is equipped with several processors working in parallel, to manage the relative measurement to each neighbor. For the measurement between i and j, node i has d processors. However, the measurement is available after dT time, introducing a delay of d iterations.

Note that for both scenarios, it would be possible to consider larger periods for every iteration, i.e., time periods dT instead of T, and using classical dynamic consensus strategies. However, as we show later in the simulations (Section 6.3 and Fig. 6), this would imply sampling the reference signals at a lower rate, giving rise to worse results for the dynamic consensus case. This problem, which does not appear in static scenarios, is challenging for the dynamic consensus case.

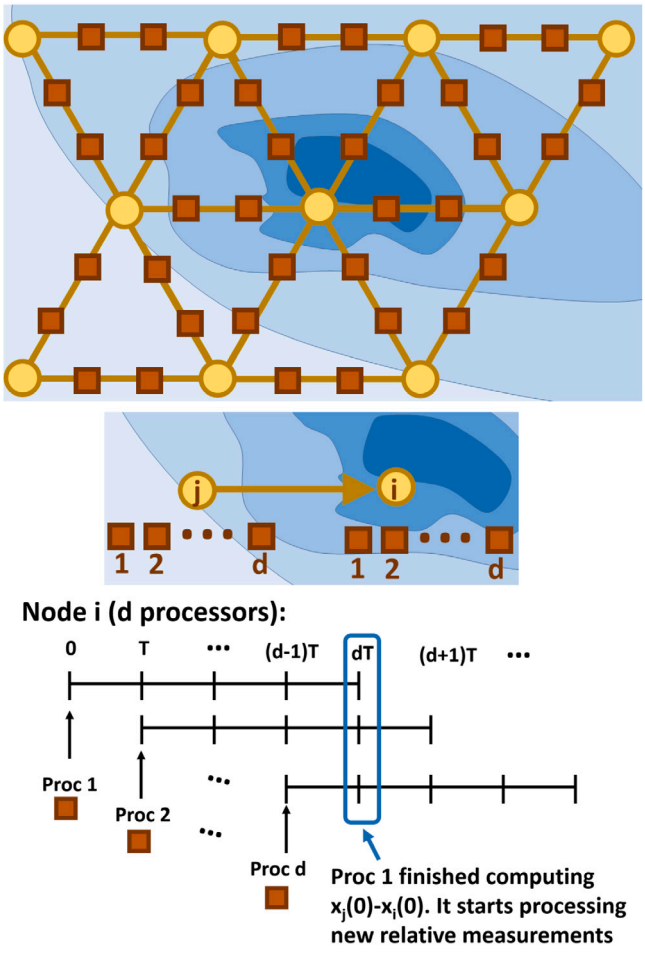


Fig. 1. Examples of application to illustrate the interest of the proposed scenario.

3. Dynamic average consensus with delay compensation

This section presents the proposed dynamic average consensus with delay compensation method. Every agent, for $i \in \{1,\dots,n\}$, runs:

$$
\begin{aligned}
&x_i(k) = x_i(k-1) + u_i(k), \ \text{for } k \geq 1,\\
&u_i(k) = r_i(k) - (1-\gamma K)\, r_i(k-1) - \gamma K r_i(k-1-d)\\
&\quad - K \sum_{j=1}^{n} a_{ij} \left(x_i(k-1-d) - x_j(k-1-d)\right)\\
&\quad - \gamma K \sum_{f=1}^{d} u_i(k-f),\\
&x_i(k') = r_i(k') \text{ for } k' \in \{-d,\dots,0\},\\
&u_i(k') = r_i(k') - r_i(k'-1) \text{ for } k' \in \{1-d,\dots,0\},
\end{aligned}
\tag{10}
$$

where d is the communication delay defined in Assumption 1, γ, K, are the controller parameters whose tuning is discussed later in Section 5, $r_i(k)$ is the reference measured by agent i at step k, and $x_i(k)$ is the state of agent i to track the average of the references at step k. Note that the state $x_i(k)$ is computed using the current measurement of the reference signal $r_i(k)$ (it appears inside the expression of $u_i(k)$ in (10)).

We note the following characteristics of the proposed method (10). First, note that both the proposed algorithm (10) and the classical dynamic average method (7) only require exchanging a piece of data of constant size with neighbors (recall from (1) that $a_{ij} = 1$ only for neighbor agents). Thus, both (7) and (10) have good scalability properties on the number of agents n in what refers to the communication costs, i.e., the costs do not increase for networks with more sensing agents. Note also that the operations carried out by each agent are light (multiplications and sums of elements of constant size) so the computational complexity is low and scalable as well. Now, consider the memory costs. For conventional consensus methods (7), memory costs are constant. The proposed method (10) requires maintaining an up-to-date sum of the last d control inputs $u_i(k-d)+\cdots+u_i(k-1)$, and also recalling the value of the reference variable $r_i(k-1-d)$ measured $d+1$ steps earlier, where d is the communication delay. Thus, the memory cost now is larger than for (7). However, the memory cost

does not depend on the number of robots n so the proposed method is scalable. In addition, all the required data is local to agent i and thus, the communication cost is not increased. Finally, from the discussion in Section 2.1, note that (10) allows the delay d to affect the relative data from neighboring robots $\big(x_i(k-1-d)-x_j(k-1-d)\big)$.

Algorithm 1 shows the pseudocode version of (10), i.e., the instructions that are run by every agent i at every step k.

Algorithm 1 *Dynamic Average Consensus with Delay Compensation* - Agent i, step k

```
Measure r_i(k)
if step k ≤ 0 then:
    u_i(k) := r_i(k) − r_i(k − 1)
    x_i(k) := r_i(k)
else
    Receive relative states from neighbors j
        (delayed data): (x_i(k − 1 − d) − x_j(k − 1 − d))
    Compute u_i(k), x_i(k) with (10)
    Discard outdated data (r_i(k − 1 − d) and u_i(k − d))
end if
Send state x_i(k) to neighbors
Store local data r_i(k), u_i(k)
```

Due to the decentralized nature of the control scheme, the algorithm in (10) cannot achieve an exact delay compensation in general. However, as we will prove in the remainder of the manuscript, the proposed method (10) effectively counteracts the negative effects of the time delays. Under an appropriate design of the controller parameters, the method can resist arbitrarily large delays without jeopardizing the closed-loop stability.

Remark 1. The control design parameters associated with the dynamic consensus with delay compensation (10), are K and γ. In Theorem 1 and Corollary 1 (Section 5), we give analytical expressions to compute such parameters for the case of general strongly connected graphs and undirected graphs, respectively. These expressions depend exclusively on the network topology, in particular on the eigenvalues of the graph Laplacian. These eigenvalues can be obtained by the robots in several ways, for instance, they can be computed in a distributed fashion. Examples include [46] for undirected graphs, and [47] for digraphs.

Remark 2. From the previous discussion, we summarize the features of the proposed dynamic consensus with delay compensation method (10). It has communication and computational costs which are constant per iteration and agent, and memory costs proportional to the delay d. The method is scalable: the communication, computational, and memory costs do not depend on the number of agents n. The control design parameters K and γ have closed-form expressions (Theorem 1 and Corollary 1) oriented towards improving the convergence speed.

4. Augmented closed-loop representation

In the previous section, we presented the proposed algorithm for dynamic average consensus with delay compensation (10). In this section, we present an equivalent augmented state-space representation of the overall multi-agent system (later in Proposition 1) that will be useful to address the convergence analysis carried out later in Section 5.

4.1. Compact form of the proposed algorithm

First, define the compact vectors $\mathbf{x}(k)\in\mathbb{R}^n$, $\mathbf{r}(k)\in\mathbb{R}^n$, whose entries contain the state variables and reference signals for all the robots in the network at step k,

$$\mathbf{x}(k)=\big[x_1(k),\dots,x_n(k)\big]^T,$$

$$\mathbf{r}(k)=\big[r_1(k),\dots,r_n(k)\big]^T. \tag{11}$$

First, we present the following two lemmas required to obtain the state-space model given later in Proposition 1:

Lemma 1 (*Compact Form*). *Let vectors $\mathbf{x}(k)$ and $\mathbf{r}(k)$ be as in (11) and let $\mathcal{L}$ be the Laplacian matrix associated with the graph (1). The dynamic average consensus algorithm with delay compensation in (10) in compact form is given by*

$$\begin{aligned}\mathbf{x}(k)&=(1-\gamma K)\,\mathbf{x}(k-1)+K\,(\gamma\mathbf{I}-\mathcal{L})\,\mathbf{x}(k-1-d)\\&\quad+\mathbf{r}(k)-(1-\gamma K)\,\mathbf{r}(k-1)-\gamma K\mathbf{r}(k-1-d),\ \text{for } k\geq 1,\\ \mathbf{x}(k')&=\mathbf{r}(k'),\ \text{for } k'\in\{-d,\dots,0\},\end{aligned} \tag{12}$$

where d is the communication delay and scalars γ and K are the control parameters defined in (10) that can be tuned to achieve a desired behavior.

Proof. First, note from (10) that

$$u_i(k)=x_i(k)-x_i(k-1), \tag{13}$$

so that the term with the sum of control inputs along steps gives

$$\begin{aligned}&\sum_{f=1}^{d}u_i(k-f)=u_i(k-1)+u_i(k-2)+\cdots+u_i(k-d)\\&=(x_i(k-1)-x_i(k-2))+(x_i(k-2)-x_i(k-3))\\&+\cdots+(x_i(k-d)-x_i(k-d-1))\\&=x_i(k-1)-x_i(k-1-d).\end{aligned} \tag{14}$$

Thus, $u_i(k)$ in (10) gives

$$\begin{aligned}u_i(k)&=-K\sum_{j=1}^{n}a_{ij}\left(x_i(k-1-d)-x_j(k-1-d)\right)\\&\quad-\gamma K x_i(k-1)+\gamma K x_i(k-1-d)\\&\quad+r_i(k)-(1-\gamma K)\,r_i(k-1)-\gamma K r_i(k-1-d).\end{aligned} \tag{15}$$

Using (15) in (10), $x_i(k)$ evolves according to

$$\begin{aligned}x_i(k)&=-K\sum_{j=1}^{n}a_{ij}\left(x_i(k-1-d)-x_j(k-1-d)\right)\\&\quad+x_i(k-1)-\gamma K x_i(k-1)+\gamma K x_i(k-1-d)\\&\quad+r_i(k)-(1-\gamma K)\,r_i(k-1)-\gamma K r_i(k-1-d),\ \text{for } k\geq 1,\\ x_i(k')&=r_i(k')\ \text{for } k'\in\{-d,\dots,0\}.\end{aligned} \tag{16}$$

Rearranging terms in (16) and expressing them using the vectors in (11), we get (12), concluding the proof.

Let $\mathbf{e}(k)\in\mathbb{R}^n$, $\bar{\mathbf{e}}(k)\in\mathbb{R}^n$ be the vectors containing respectively the error vector, and the average error vector, at every step k,

$$\begin{aligned}\mathbf{e}(k)&=\mathbf{x}(k)-\frac{\mathbf{1}w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}\mathbf{r}(k),\\ \bar{\mathbf{e}}(k)&=\frac{\mathbf{1}w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}\mathbf{e}(k)=\frac{\mathbf{1}w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}\mathbf{x}(k)-\frac{\mathbf{1}w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}\mathbf{r}(k),\end{aligned} \tag{17}$$

where $w_{\mathcal{L}_1}$ is the left eigenvector of the Laplacian matrix (1) associated with the eigenvalue 0 defined in (2), and where $\mathbf{x}(k)$ and $\mathbf{r}(k)$ are defined in (11). Note from the equivalence $w_{\mathcal{L}_1}^T\mathbf{1}=\mathbf{1}^T w_{\mathcal{L}_1}$ that $\frac{\mathbf{1}(w_{\mathcal{L}_1}^T\mathbf{1})w_{\mathcal{L}_1}^T}{(\mathbf{1}^T w_{\mathcal{L}_1})(\mathbf{1}^T w_{\mathcal{L}_1})}=\frac{\mathbf{1}w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}$.

Lemma 2 (*Error and Average Error Vectors*). *The dynamic average consensus with delay compensation method in (10) and (12), expressed in terms of the error vector $\mathbf{e}(k)$ in (17), under Assumption 2 leads to the following closed-loop expressions:*

$$\mathbf{e}(k)=(1-\gamma K)\,\mathbf{e}(k-1)+K\,(\gamma\mathbf{I}-\mathcal{L})\,\mathbf{e}(k-1-d)+\Pi\mathbf{r}(k)$$

$$- (1-\gamma K)\,\Pi\mathbf{r}(k-1) - \gamma K \Pi \mathbf{r}(k-1-d), \text{ for } k \geq 1,$$

$$\mathbf{e}(k') = \Pi \mathbf{r}(k'), \text{ for } k' \in \{-d, \dots, 0\}, \tag{18}$$

where $\mathcal{L}$ is the Laplacian matrix in (1), and matrix $\Pi \in \mathbb{R}^{n\times n}$ is given by

$$\Pi = \mathbf{I} - \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}. \tag{19}$$

Moreover, the average error vector $\bar{\mathbf{e}}(k)$ in (17), whose dynamics error $\mathbf{e}(k)$ is given in (18), is identically zero, i.e.,

$$\bar{\mathbf{e}}(k) = \mathbf{0}_n, \forall k \geq 0. \tag{20}$$

Proof. Expression (18) is obtained by applying the change of variables (17) in (12), and using $\mathcal{L}\mathbf{1} = \mathbf{0}$.

In order to prove (20), we express (18) in terms of the average error vector $\bar{\mathbf{e}}(k)$ given in (17), obtaining

$$\begin{aligned}\bar{\mathbf{e}}(k) = (1-\gamma K)\bar{\mathbf{e}}(k-1) + \left(K\gamma \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}\right)\mathbf{e}(k-1-d) \\ - \left(K \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \mathcal{L}\right)\mathbf{e}(k-1-d) + \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \Pi\,(\mathbf{r}(k) - \mathbf{r}(k-1)) \\ + \gamma K \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \Pi\,(\mathbf{r}(k-1) - \mathbf{r}(k-1-d)), \text{ for } k \geq 1 \text{ and}\end{aligned}$$

$$\bar{\mathbf{e}}(k') = \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \Pi \mathbf{r}(k'), \text{ for } k' \in \{-d, 0\}. \tag{21}$$

Note that

$$\frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \mathcal{L} = \mathbf{0}, \text{ and } \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \Pi = \mathbf{0}. \tag{22}$$

Using (22) in (21), several terms cancel out, and we obtain

$$\bar{\mathbf{e}}(k) = (1-\gamma K)\bar{\mathbf{e}}(k-1) + K\gamma \bar{\mathbf{e}}(k-1-d), \text{ for } k \geq 1 \text{ and}$$

$$\bar{\mathbf{e}}(k') = \mathbf{0}_n, \text{ for all } k' \in \{-d, 0\}. \tag{23}$$

Finally, the condition (20) can be straightforwardly deduced from (23), concluding the proof.

Observe from (23) that $\bar{\mathbf{e}}(k)$ starts being equal to zero, and continues being zero for all iterations k.

4.2. Augmented closed-loop model

This section finds the augmented system model formed by gathering current and past states, errors, and references, with a horizon equivalent to delay d.

Definition 1 (*Augmented Vectors*)**.** We define the following augmented vectors $\mathbf{x}^+(k) \in \mathbb{R}^{n(d+1)}$, $\mathbf{e}^+(k) \in \mathbb{R}^{n(d+1)}$, $\mathbf{r}^+(k) \in \mathbb{R}^{n(d+2)}$,

$$\begin{aligned}\mathbf{x}^+(k) &= \left[\mathbf{x}(k)^T, \mathbf{x}(k-1)^T, \dots, \mathbf{x}(k-d)^T\right]^T, \\ \mathbf{e}^+(k) &= \left[\mathbf{e}(k)^T, \mathbf{e}(k-1)^T, \dots, \mathbf{e}(k-d)^T\right]^T, \\ \mathbf{r}^+(k) &= \left[\mathbf{r}(k)^T, \mathbf{r}(k-1)^T, \dots, \mathbf{r}(k-1-d)^T\right]^T,\end{aligned} \tag{24}$$

with $\mathbf{x}(k)$, $\mathbf{e}(k)$, and $\mathbf{r}(k)$ as in Eqs. (11), (17). Note that the augmented reference vector $\mathbf{r}^+(k) \in \mathbb{R}^{n(d+2)}$ contains an additional element due to the inclusion of the reference $r_i(k-1-d)$ at step $k-1-d$.

The following result obtains an equivalent closed-loop representation of the multi-agent system with the proposed dynamic average consensus with delay compensation (10).

Proposition 1 (*Augmented Form*)**.** *Let $\mathbf{x}^+(k)$, $\mathbf{e}^+(k)$ and $\mathbf{r}^+(k)$ be the vectors defined in (24) and let $\mathcal{L}$ be the Laplacian matrix associated with the graph (1). Let scalars γ, K be the controller parameters defined in (10). The dynamic average consensus algorithm with delay compensation in (10), (12), (18), in augmented form, for all $k \geq 1$, is given by*

$$\begin{aligned}\mathbf{x}^+(k) &= M\mathbf{x}^+(k-1) + B\mathbf{r}^+(k), \\ \mathbf{e}^+(k) &= M\mathbf{e}^+(k-1) + B_e\mathbf{r}^+(k), \text{ where}\end{aligned} \tag{25}$$

$$M = \begin{bmatrix} (1-\gamma K)\mathbf{I}_n & \mathbf{0}_{n\times(d-1)n} & \gamma K\mathbf{I}_n - K\mathcal{L} \\ \mathbf{I}_n & \mathbf{0}_{n\times(d-1)n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{(d-1)n\times n} & \mathbf{I}_{(d-1)n} & \mathbf{0}_{(d-1)n\times n} \end{bmatrix},$$

$$B = \begin{bmatrix} \mathbf{I}_n & -(1-\gamma K)\mathbf{I}_n & \mathbf{0}_{n\times n(d-1)} & -\gamma K\mathbf{I}_n \\ & & \mathbf{0}_{n(d+1)\times n(d+2)} & \end{bmatrix},$$

$$B_e = \begin{bmatrix} \Pi & -(1-\gamma K)\Pi & \mathbf{0}_{n\times n(d-1)} & -\gamma K\Pi \\ & & \mathbf{0}_{n(d+1)\times n(d+2)} & \end{bmatrix},$$

where n is the number of robots, d is the communication delay, and Π is given in (19).

Proof. It follows from Lemmas 1 and 2, and the definition of the augmented vectors (24).

5. Convergence analysis based on eigenvalue approach

5.1. Eigenvalues characterization

Proposition 2 (*Eigenvalues and Eigenvectors*)**.** *Let $\lambda_{\mathcal{L}_i}$ and $w_{\mathcal{L}_i}$ be the eigenvalues and left eigenvectors of the Laplacian matrix in (1), and let Assumption 2 hold. The eigenvalues $\lambda_i \in \mathbb{C}$ and the left eigenvectors $\mathbf{w}_i \in \mathbb{C}^{(d+1)n}$ (the entries may be complex values) of the augmented system matrix M in (25) satisfy the following expressions, for $i \in \{1, \dots, n\}$:*

$$\begin{aligned}\lambda_1^{d+1} - \lambda_1^d(1-\gamma K) - \gamma K = 0, \\ \lambda_2^{d+1} - \lambda_2^d(1-\gamma K) + (K\lambda_{\mathcal{L}_2} - \gamma K) = 0, \\ \vdots \\ \lambda_n^{d+1} - \lambda_n^d(1-\gamma K) + (K\lambda_{\mathcal{L}_n} - \gamma K) = 0,\end{aligned} \tag{26}$$

where λ_i^d represents the value of λ_i raised to the dth power. Each line in (26) gives $d+1$ solutions for λ_i, and

$$\mathbf{w}_i = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ w_{i,3} \\ \vdots \\ w_{i,d+1} \end{bmatrix} = \begin{bmatrix} w_{\mathcal{L}_i} \\ -(1-\gamma K - \lambda_i) w_{\mathcal{L}_i} \\ -\lambda_i(1-\gamma K - \lambda_i) w_{\mathcal{L}_i} \\ \vdots \\ -(\lambda_i)^{(d-1)}(1-\gamma K - \lambda_i) w_{\mathcal{L}_i} \end{bmatrix}, \tag{27}$$

where every $w_{i,l}$, for $l = 1, \dots, d+1$ contains n elements.

Proof. Recall from Proposition 1 that the system matrix M is given by (25). Now we consider the left eigenvectors $\mathbf{w}_i \in \mathbb{C}^{(d+1)n}$ and eigenvalues λ_i of M, for $i = 1, \dots, n$.

Considering that

$$\left[w_{i,1}^T, \dots, w_{i,d+1}^T\right] M = \lambda_i \left[w_{i,1}^T, \dots, w_{i,d+1}^T\right], \tag{28}$$

we get the following system

$$\begin{aligned}(1-\gamma K) w_{i,1} + w_{i,2} &= \lambda_i w_{i,1} \\ w_{i,3} &= \lambda_i w_{i,2} \\ &\vdots \\ w_{i,d+1} &= \lambda_i w_{i,d} \\ w_{i,1}^T(\gamma K - K\mathcal{L}) &= \lambda w_{i,d+1}^T.\end{aligned} \tag{29}$$

We let $w_{i,1}$ be a left eigenvector of the Laplacian matrix, $w_{i,1} = w_{\mathcal{L}_i}$ (27), so that

$$w_{i,1}^T K\mathcal{L} = \lambda_{\mathcal{L}_i} K w_{i,1}^T, \tag{30}$$

and thus the last expression in (29) gives

$$(\gamma K - K\lambda_{\mathcal{L}_i}) w_{i,1} = \lambda_i w_{i,d+1} = \lambda_i^2 w_{i,d} = \dots = \lambda_i^d w_{i,2}, \tag{31}$$

where

$$w_{i,2} = (\lambda_i - (1-\gamma K))w_{i,1}, \quad \text{so that}$$
$$\lambda_i^d w_{i,2} = \lambda_i^{d+1} w_{i,1} - \lambda_i^d (1-\gamma K) w_{i,1}, \tag{32}$$

giving finally

$$(\gamma K - K\lambda_{\mathcal{L}_i}) w_{i,1} = \lambda_i^{d+1} w_{i,1} - \lambda_i^d (1-\gamma K) w_{i,1}. \tag{33}$$

Thus, we conclude that the eigenvalues λ of the augmented system matrix M are the ones satisfying (26). The expression in (27) is obtained by making $w_{i,1} = w_{\mathcal{L}_i}$ and applying (29).

Observe from Proposition 2 that the system matrix M in (25) associated with the augmented system has one eigenvalue equal to 1, and d additional eigenvalues associated with the Laplacian eigenvalue $\lambda_{\mathcal{L}_1} = 0$, i.e., the $d+1$ solutions of:

$$\lambda_1^{d+1} - \lambda_1^d (1-\gamma K) - \gamma K = 0, \tag{34}$$

where d is the time delay expressed in time steps. From Proposition 2 and its proof, the associated eigenvectors are

$$\left(c_1 \frac{w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}, c_2 \frac{w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}}, \dots, c_{d+1} \frac{w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \right)^T, \tag{35}$$

where $c_1, \dots, c_{d+1}$ are real or complex constants, and the left Laplacian eigenvector $w_{\mathcal{L}_1}^T$ satisfies (22).

We let $\lambda_{M,1} \in \mathbb{C}^{(d+1)\times(d+1)}$ be a diagonal matrix with these eigenvalues (the solutions of (34)), and we let $W_{M,1}^l \in \mathbb{C}^{(d+1)n\times(d+1)}$ contain the associated $(d+1)$ left eigenvectors of M, as in (35). Equivalently, we let $W_{M,1}^r$ be the associated right eigenvectors; although their expressions could be obtained in a similar way to $W_{M,1}^l$ in Proposition 2, this is omitted since it is not needed in the following result. Observe that matrix

$$M_1 = M - W_{M,1}^r \lambda_{M,1} (W_{M,1}^l)^T, \tag{36}$$

has the same eigenvalues as matrix M, except for the eigenvalues in $\lambda_{M,1}$ that appeared in M but now have value equal to 0 in the new matrix M_1.

Proposition 3. *The dynamic average consensus with delay compensation (10) under Assumptions 1, 2, converges to a neighborhood of the weighted average (2) of the reference signals, i.e.,*

$$\limsup_{k\to\infty} \left| x_i(k) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^n [w_{\mathcal{L}_1}]_j r_j(k) \right| \le \mathcal{B}, \tag{37}$$

for all $i \in \{1, \dots, n\}$, with $\mathcal{B} < \infty$, if the spectral radius of matrix M_1 is less than 1, where M_1 is given by (36), and it has the same eigenvalues as the augmented system matrix M in (25), but considering exclusively the ones with $i \in \{2, \dots, n\}$, i.e., excluding the $d+1$ eigenvalues with $i = 1$ (recall that matrices M and M_1 have $(d+1)n$ eigenvalues), and if Assumption 3 holds with α_∞^+ finite and defined as follows, $\forall k$:

$$\| \Pi(\mathbf{r}(k) - \mathbf{r}(k-1)) + \gamma K \Pi(\mathbf{r}(k-1) - \mathbf{r}(k-1-d)) \|_\infty \le \alpha_\infty^+. \tag{38}$$

Moreover, the convergence speed of (10) can be determined by the spectral radius of M_1 (namely $\rho(M_1)$).

Proof. Recall from (25) that, for all $k \ge 1$,

$$\mathbf{e}^+(k) = M\mathbf{e}^+(k-1) + B_e \mathbf{r}^+(k). \tag{39}$$

Now we rewrite (39) in terms of matrix M_1 in (36):

$$\mathbf{e}^+(k) = M_1 \mathbf{e}^+(k-1) + B_e \mathbf{r}^+(k) + W_{M,1}^r \lambda_{M,1} (W_{M,1}^l)^T \mathbf{e}^+(k-1), \text{ for } k \ge 1. \tag{40}$$

Taking into account the definition of $\bar{\mathbf{e}}(k)$ in (17) and (20) in Lemma 2, together with the structure of the eigenvectors given in (35), and considering that the augmented error vector $\mathbf{e}^+(k)$ is composed of error vectors at different time steps (see Proposition 1), the last term in (40) vanishes. Thus, we get that the errors evolve according to

$$\mathbf{e}^+(k) = M_1 \mathbf{e}^+(k-1) + B_e \mathbf{r}^+(k), \tag{41}$$

so that the stability and convergence speed depends on the additional eigenvalues of matrix M, i.e., the ones associated with the Laplacian eigenvalues $\lambda_{\mathcal{L}_2}, \dots, \lambda_{\mathcal{L}_n}$.

Now, we decompose matrix M_1 into its eigenvalues λ_{M_1} and right $W_{M_1}^r$ and left $W_{M_1}^l$ eigenvectors,

$$M_1 = W_{M_1}^r \lambda_{M_1} (W_{M_1}^l)^T, \tag{42}$$

with $W_{M_1}^r (W_{M_1}^l)^T = \mathbf{I}$, and we let $\rho(M_1)$ be the largest modulus eigenvalue of M_1. Then, from (41) and (42),

$$\mathbf{e}^+(k) = W_{M_1}^r (\lambda_{M_1})^k (W_{M_1}^l)^T \mathbf{e}^+(0) + \sum_{f=0}^{k-1} W_{M_1}^r (\lambda_{M_1})^f (W_{M_1}^l)^T B_e \mathbf{r}^+(k-f). \tag{43}$$

Note that, since λ_{M_1} is a diagonal matrix, then

$$\|(\lambda_{M_1})^k\|_\infty \le \rho(M_1)^k. \tag{44}$$

From (43), (44) and (46), we have

$$\|\mathbf{e}^+(k)\|_\infty \le \mu_1\mu_2\rho(M_1)^k \|\mathbf{e}^+(0)\|_\infty + \sum_{f=0}^{k-1} \mu_1\mu_2 (\rho(M_1))^f \|B_e \mathbf{r}^+(k-f)\|_\infty. \tag{45}$$

where μ_1 and μ_2 are defined as:

$$\mu_1 = \|W_{M_1}^l\|_\infty, \qquad \mu_2 = \|W_{M_1}^r\|_\infty. \tag{46}$$

Note from Proposition 1 that

$$B_e \mathbf{r}^+(k) = \Pi(\mathbf{r}(k) - \mathbf{r}(k-1)) + \gamma K \Pi(\mathbf{r}(k-1) - \mathbf{r}(k-1-d)). \tag{47}$$

Consider Assumption 3 with the bound α_∞^+ defined in (38). Then,

$$\|B_e \mathbf{r}^+(k')\|_\infty \le \alpha_\infty^+, \forall k', \tag{48}$$

with α_∞^+ finite. Thus, (45) gives

$$\|\mathbf{e}^+(k)\|_\infty \le \mu_1\mu_2\rho(M_1)^k \|\mathbf{e}^+(0)\|_\infty + \mu_1\mu_2 \left(\sum_{f=0}^{k-1} \rho(M_1)^f \right) \alpha_\infty^+ \le \mu_1\mu_2\rho(M_1)^k \|\mathbf{e}^+(0)\|_\infty + \mu_1\mu_2 \frac{1-\rho(M_1)^k}{(1-\rho(M_1))} \alpha_\infty^+. \tag{49}$$

Now, note that

$$\left| x_i(k) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^n [w_{\mathcal{L}_1}]_j r_j(k) \right| \le \|\mathbf{e}^+(k)\|_\infty,$$
$$\limsup_{k\to\infty} \left| x_i(k) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^n [w_{\mathcal{L}_1}]_j r_j(k) \right| \le \limsup_{k\to\infty} \|\mathbf{e}^+(k)\|_\infty. \tag{50}$$

If $\rho(M_1) \in [0, 1)$, then $1 - \rho(M_1) \in (0, 1]$ and

$$\lim_{k\to\infty} \rho(M_1)^k = 0, \tag{51}$$

so that, from (49),

$$\limsup_{k\to\infty} \|\mathbf{e}^+(k)\|_\infty \le \mu_1\mu_2 \frac{1}{(1-\rho(M_1))} \alpha_\infty^+. \tag{52}$$

Recall from Assumption 3 that α_∞^+ is finite. From (50) and (52), we get (37), with

$$\mathcal{B} = \mu_1\mu_2 \frac{1}{(1-\rho(M_1))} \alpha_\infty^+, \tag{53}$$

which concludes the proof.

From the previous analysis, it can be seen that the proposed method (10) can be studied by using the classical tools for analyzing consensus methods, using the augmented system in (25) instead of, e.g., the matrix form of (4) that includes the Perron matrix $\mathbf{I}_n - \beta\delta\mathcal{L}$. Thus, following similar ideas as in [1,2], we built an expression in Proposition 3 for the error bound similar to (6). Note that the bound given by (37) and (53) is quite conservative. In conclusion, we have shown that the ultimate tracking error *is bounded* under similar conditions as the ones considered in classical dynamic consensus methods [1,2], i.e., as long as the variation of the reference signals (relative to their centroid) along the iterations is bounded, i.e., when there exists a finite α_∞^+ as in Assumption 3.

5.2. Root locus analysis

Note from Propositions 2 and 3 that the stability and convergence speed of the proposed method (10) can be studied by considering the eigenvalues of the augmented system matrix M (25), excluding the $d+1$ eigenvalues of M associated with the Laplacian eigenvalue $\lambda_{\mathcal{L}_1} = 0$. Thus, next, we discuss the values of these remaining eigenvalues, associated with the augmented closed-loop system (25), depending on the parameter selection. In the following discussion, we analyze the effects of the parameters for the real eigenvalues. The case in which the eigenvalues are complex conjugates would be done in a similar way, and it is omitted for clarity. We let $\lambda_{\mathcal{L}_i}$ be the ith eigenvalue of the Laplacian, and λ_i be the eigenvalue of the resulting system matrix M of the delayed system, for $i = 2, \dots, n$. From Proposition 2:

$$\begin{aligned}
&\lambda_1^{d+1} - a\lambda_1^d + \Omega_1 = 0\\
&\lambda_2^{d+1} - a\lambda_2^d + \Omega_2 = 0\\
&\qquad\vdots\\
&\lambda_n^{d+1} - a\lambda_n^d + \Omega_n = 0
\end{aligned} \tag{54}$$

where each line gives $d+1$ values for λ_i, and a, Ω_i, $i = 1, \dots, n$ are defined as:

$$a = 1 - \gamma K, \qquad \Omega_i = (K\lambda_{\mathcal{L}_i} - \gamma K). \tag{55}$$

Observe in (54) that, in all cases, coefficient $a = (1 - \gamma K)$ is the same, whereas coefficient Ω_i changes depending on the particular Laplacian eigenvalue. Now, we discuss how the eigenvalues change depending on parameters a and Ω_i, whose values depend on the tuning parameters γ, K, and on the eigenvalues of the Laplacian matrix $\lambda_{\mathcal{L}_i}$.

For our analysis, we resort to root locus analysis, as in [48]. Whereas in [48] it was used to analyze general linear systems with delays and *uni-dimensional* variables, here apply the root locus analysis to study the multi-dimensional dynamic consensus method presented previously. In terms of root locus analysis, each row in (54) is equivalent to having a system in open loop, $\mathcal{F}_{open-loop}(\lambda)$, with

$$\mathcal{F}_{open-loop}(\lambda) = \frac{1}{\lambda^{d+1} - a\lambda^d} = \frac{1}{\lambda^d(\lambda - a)}, \tag{56}$$

i.e., in open loop, there are d poles in $\lambda = 0$, and one pole in $\lambda = a$. Then, the root locus method can be used to study the placement of the closed-loop poles (the solutions of (54)) depending on the specific values of the Ω_i, for $i = 2, \dots, n$. Thus, the shape of the root locus is the same for all the equations in (54), whereas every particular value associated with Ω_i will fix the particular solutions.

The breakaway point for the root locus is computed by making

$$\frac{\partial \mathcal{F}_{open-loop}(\lambda)}{\partial \lambda} = 0, \tag{57}$$

and it gives

$$\lambda_{break} = \frac{da}{d+1}, \qquad \Omega_{break} = \frac{d^d}{(d+1)^{(d+1)}}(|a|)^{d+1}, \tag{58}$$

where d is the time delay expressed in time steps, and d^d represents the value of the scalar d raised to the dth power (equivalently, $(d+1)^{(d+1)}$ represents $d+1$ raised to the $(d+1)$-th power). Depending on the particular sign of parameters a and Ω_i, there may not be any breakaway point. We discuss these situations in Fig. 2.

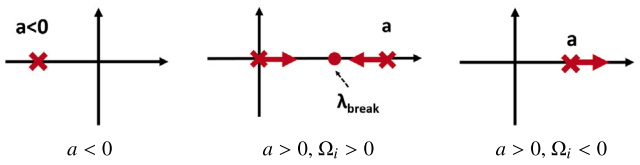


Fig. 2. General shape of the root locus.

Note that if $a < 0$ (Fig. 2, left), then there is an open loop pole with a negative sign. This case is symmetric to $a > 0$, with the disadvantage that these poles with negative signs are known to produce ringing behavior [49]. Thus, we keep the parameter a positive, as in Fig. 2 (center and rightmost-side pictures),

$$a \geq 0, \qquad \gamma K \leq 1. \tag{59}$$

Note that in Fig. 2 Right, parameter $\Omega_i < 0$ is negative. Thus, as the modulus of Ω_i increases, the pole that starts at a becomes larger (slower, or even unstable). On the other hand, in Fig. 2 Center ($a > 0$, $\Omega_i > 0$), this pole becomes smaller (faster, more stable) as Ω_i increases. For values between a and the breakaway point λ_{break} (58), it takes real values, leading to an overdamped behavior. Moreover, the fastest response is obtained when the pole is equal to λ_{break}.

From the above discussion, it can be seen that the speed of convergence (the largest modulus closed-loop eigenvalue) cannot be faster than $\lambda_{break} = \frac{da}{d+1}$. Thus, having a smaller a will give rise to a faster convergence speed. This is especially important for large delays since $\frac{d}{d+1}$ becomes closer to one as d increases. Thus, in this paper, we propose the following tuning method.

Parameter tuning: The following criteria are proposed for choosing the controller parameter γ of the dynamic average consensus with delay compensation (10):

$$\gamma = \frac{1}{K}, \text{ i.e., } \quad \gamma K = 1 \text{ so that } a = 0. \tag{60}$$

As shown later in Theorem 1, the above equivalence ensures the convergence of the overall system, irrespective of the size of delay d, i.e., the system matrix is stable for all values of the delay d, which is an important characteristic. Note also that the parameter selection for $K_\star$ and $\gamma_\star$ in Theorem 1 does not depend on the delay d.

Definition 2 (*Laplacian Eigenvalues*)**.** Under Assumption 2, the Laplacian has a single trivial eigenvalue $\lambda_{\mathcal{L}_1} = 0$. For undirected graphs, all the remaining eigenvalues are real and strictly positive [25], and we let $\lambda_{\mathcal{L}_2}$ and $\lambda_{\mathcal{L}_n}$ be the minimum and maximum eigenvalues, excluding the trivial eigenvalue $\lambda_{\mathcal{L}_1} = 0$. For graphs that are strongly connected [25, Lemma 2], the eigenvalues may be real or complex conjugates; all non-trivial eigenvalues satisfy $\cos(\lambda_{\mathcal{L}_i}) \in (0, 1]$, for $i \in \{2, \dots, n\}$. We let $\lambda_{\mathcal{L}_{\cos}}$ be the minimum cosine, and $\lambda_{\mathcal{L}_{\min}}$ and $\lambda_{\mathcal{L}_{\max}}$ be the smallest and largest modulus Laplacian eigenvalues, excluding the trivial eigenvalue:

$$\begin{aligned}
&\lambda_{\mathcal{L}_{\cos}} = \min_{i\in\{2,\dots,n\}} \cos(\lambda_{\mathcal{L}_i}),\\
&\lambda_{\mathcal{L}_{\min}} = \min_{i\in\{2,\dots,n\}} |\lambda_{\mathcal{L}_i}|, \ \lambda_{\mathcal{L}_{\max}} = \max_{i\in\{2,\dots,n\}} |\lambda_{\mathcal{L}_i}|.
\end{aligned} \tag{61}$$

Theorem 1 (*Stability and Convergence Speed*)**.** *Consider robots run the dynamic consensus with delay compensation (10), where there is a delay of d steps associated with the communication, and Assumption 1 holds. Let the directed communication graph be as in Assumption 2, with associated Laplacian matrix $\mathcal{L}$. Let parameters $\gamma = \gamma_\star$, $K = K_\star$ satisfy*

$$K_\star = \frac{2\lambda_{\mathcal{L}_{\cos}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})}, \qquad \gamma_\star = \frac{1}{K_\star}, \tag{62}$$

where $\lambda_{\mathcal{L}_{\cos}}$, $\lambda_{\mathcal{L}_{\min}}$, $\lambda_{\mathcal{L}_{\max}}$ are given by (61) in Definition 2. Then, the system matrix M_1 in (36) is convergent for all values of the delay d, with convergence speed:

$$\rho(M_1) \leq \left(1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\min}} \lambda_{\mathcal{L}_{\max}}}{\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}}}\right)^{\frac{1}{2(d+1)}} < 1. \tag{63}$$

Moreover, let α_∞^+ be a finite value that bounds the variation of the reference signals as in Assumption 3 in (38) with (19):

$$\alpha_\infty^+ = \max_k \left(\left\| \left(\mathbf{I} - \frac{\mathbf{1} w_{\mathcal{L}_1}^T}{\mathbf{1}^T w_{\mathcal{L}_1}} \right) (\mathbf{r}(k) - \mathbf{r}(k-1-d)) \right\|_\infty \right). \tag{64}$$

Then, the states of the agents converge asymptotically to a neighborhood of the weighted average (2) of the reference signals, i.e., (37).

Proof. We use Lemmas 1 and 2 and Propositions 1–3 to conclude that the convergence speed of the system matrix M_1 in (36) associated with algorithm (10) can be studied by considering the eigenvalues in (26), (54), for $i = 2, \dots, n$. Note from (62) that using γ equal to $\gamma_\star = \frac{1}{K_\star}$ makes $a = 0$, with a given in (55). Then, Eq. (54) gives

$$\lambda_i^{d+1} = (1 - K\lambda_{\mathcal{L}_i}), \tag{65}$$

for all $i \in \{2, \dots, n\}$.

Now, we study the modulus of these eigenvalues. For real Laplacian eigenvalues $\lambda_{\mathcal{L}_i}$, we obtain

$$|\lambda_i|^{2(d+1)} = (1 - K\lambda_{\mathcal{L}_i})^2. \tag{66}$$

For complex conjugate Laplacian eigenvalues, $\lambda_{\mathcal{L}_i}^+$, $\lambda_{\mathcal{L}_i}^-$, (54) gives pairs of complex conjugate eigenvalues with

$$(\lambda_i^+)^{(d+1)} = (1 - K\lambda_{\mathcal{L}_i}^+), \qquad (\lambda_i^-)^{(d+1)} = (1 - K\lambda_{\mathcal{L}_i}^-), \tag{67}$$

so that the modulus of these eigenvalues satisfy

$$|\lambda_i|^{2(d+1)} = (\lambda_i^+)^{(d+1)}(\lambda_i^-)^{(d+1)} = 1 - 2K|\lambda_{\mathcal{L}_i}|\cos(\lambda_{\mathcal{L}_i}) + K^2|\lambda_{\mathcal{L}_i}|^2, \tag{68}$$

which, for real positive eigenvalues $\lambda_{\mathcal{L}_i}$, equals (66), since the cosine of real eigenvalues equals one $\cos(\lambda_{\mathcal{L}_i}) = 1$.

Observe from (62) that $K > 0$. For a fixed $K > 0$, we discuss next the expressions for the modulus $|\lambda_i|^{2(d+1)}$ in (68), depending on the values of $\cos(\lambda_{\mathcal{L}_i})$ and $|\lambda_{\mathcal{L}_i}|$.

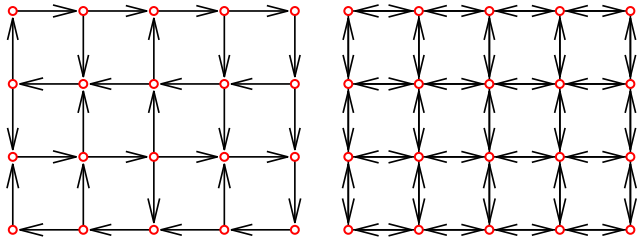
If we increase $\cos(\lambda_{\mathcal{L}_i})$ from 0 to 1, then (68) is strictly decreasing, taking always positive values and reaching its maximum for $\cos(\lambda_{\mathcal{L}_i}) = \lambda_{\mathcal{L}_{\cos}}$. Instead, if we modify the modulus $|\lambda_{\mathcal{L}_i}|$, from $\lambda_{\mathcal{L}_{\min}}$ to $\lambda_{\mathcal{L}_{\max}}$, then (68) first decreases, reaching a minimum (greater than 0), and it increases again. Thus, (68) reaches its maximum value at either $|\lambda_{\mathcal{L}_i}| = \lambda_{\mathcal{L}_{\min}}$ or $|\lambda_{\mathcal{L}_i}| = \lambda_{\mathcal{L}_{\max}}$. Thus, (68) will be strictly smaller than one for all the eigenvalues of the system matrix M_1, and thus matrix M_1 will be convergent, if (68) it is strictly smaller than one for the maximum situations detected earlier, with K as in (62).

The value of $|\lambda_i|^{2(d+1)}$ in (68) for $\cos(\lambda_{\mathcal{L}_i}) = \lambda_{\mathcal{L}_{\cos}}$ and $|\lambda_{\mathcal{L}_i}| = \lambda_{\mathcal{L}_{\min}}$, with $K = K_\star = \frac{2\lambda_{\mathcal{L}_{\cos}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})}$ as in (62), gives

$$|\lambda_i|^{2(d+1)} = 1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\min}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})} + \frac{4(\lambda_{\mathcal{L}_{\cos}})^2}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})^2}(\lambda_{\mathcal{L}_{\min}})^2 = 1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\min}} \lambda_{\mathcal{L}_{\max}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})^2} < 1, \tag{69}$$

which is strictly smaller than 1, since the second term is strictly negative. Now we consider the value of $|\lambda_i|^{2(d+1)}$ in (68) for $\cos(\lambda_{\mathcal{L}_i}) = \lambda_{\mathcal{L}_{\cos}}$ and $|\lambda_{\mathcal{L}_i}| = \lambda_{\mathcal{L}_{\max}}$, with $K = K_\star = \frac{2\lambda_{\mathcal{L}_{\cos}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})}$ as in (62), which gives

$$|\lambda_i|^{2(d+1)} = 1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\max}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})} + \frac{4(\lambda_{\mathcal{L}_{\cos}})^2}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})^2}(\lambda_{\mathcal{L}_{\max}})^2 = 1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\min}} \lambda_{\mathcal{L}_{\max}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})^2} < 1, \tag{70}$$



(a) Directed graph (b) Undirected graph

Fig. 3. Communication graphs used in the simulations. (a) Directed graph strongly connected. (b) Undirected connected graph.

as before. Thus, the system matrix M_1 is convergent, with spectral radius satisfying (63). Finally, the asymptotic convergence of the states of the agents to a neighborhood of the weighted average (2) of the reference signals, is obtained by using Proposition 3 with $\rho(M_1) < 1$, and by repeating the steps followed in the proof of Proposition 3 using $\gamma K = 1$ as in (62) and α_∞^+ as in (64). In this case, we obtain the same expression for $\mathcal{B}$ as in (53), but where α_∞^+ here is as in (64). This concludes the proof.

Corollary 1 (*Undirected Graphs*). *For undirected graphs, the expressions for the parameters $\gamma_\star$, $K_\star$ and for the convergence speed $\rho(M_1)$ are simpler, and they are given by:*

$$K_\star = \frac{2}{(\lambda_{\mathcal{L}_2} + \lambda_{\mathcal{L}_n})}, \quad \gamma_\star = \frac{1}{K_\star}, \quad \rho(M_1) = \left(\frac{\lambda_{\mathcal{L}_n} - \lambda_{\mathcal{L}_2}}{\lambda_{\mathcal{L}_n} + \lambda_{\mathcal{L}_2}}\right)^{\left(\frac{1}{d+1}\right)}, \tag{71}$$

where $\lambda_{\mathcal{L}_2}$ and $\lambda_{\mathcal{L}_n}$ are the second smallest and the largest eigenvalues of the Laplacian matrix (Definition 2).

Note that (71) requires knowing $\lambda_{\mathcal{L}_2}$ and $\lambda_{\mathcal{L}_n}$, that can be computed by the robots in a distributed way [46]. Equivalently, for directed graphs, the expressions in (62) depend on the Laplacian eigenvalues of directed graphs that can also be obtained in a distributed way [47].

Remark 3 (*Ultimate Tracking Bound*). Recall again that the bound given by (37) is quite conservative. However, the interest of the previous discussion is that we can conclude that the ultimate tracking error *is bounded* under similar conditions as the ones considered in classical dynamic consensus methods [1,2], i.e., as long as the variation of the reference signals (relative to their centroid) along the iterations is bounded.

6. Simulations

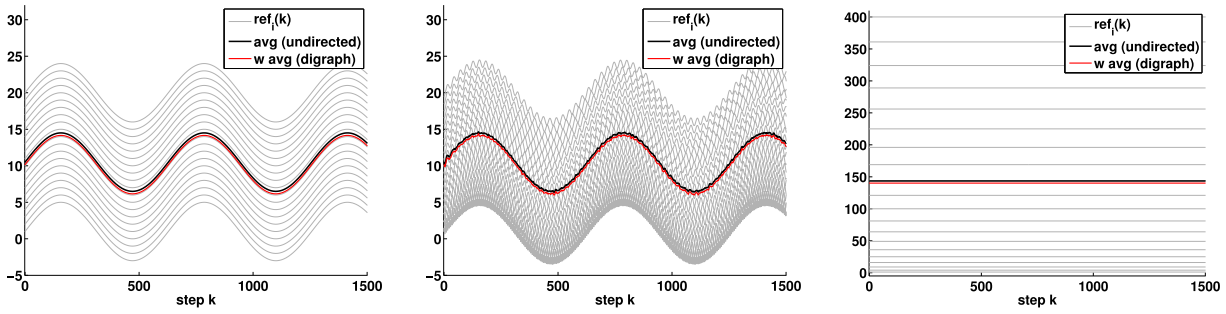
We consider an example with $n = 20$ agents tracking the average of 20 reference signals. Fig. 3 shows the communication graphs used: a directed strongly connected graph (Fig. 3(a)) and an undirected connected graph (Fig. 3(b)). We test the performance of the proposed algorithm, under different time delays ($d = \{1, 10, 50\}$), for tracking different references signals, according to three sets of simulations (Fig. 4). Next, we show the reference signal $r_i(k)$ measured by every robot $i = 1, \dots, n$ at every step $k \geq 0$:

- ($S1$) Time-varying reference signals with the same time-variation pattern.

$$r_i(k) = i + 4\sin(0.01k), \ \forall i = 1, \dots, n, \ \forall k \geq 0. \tag{72}$$

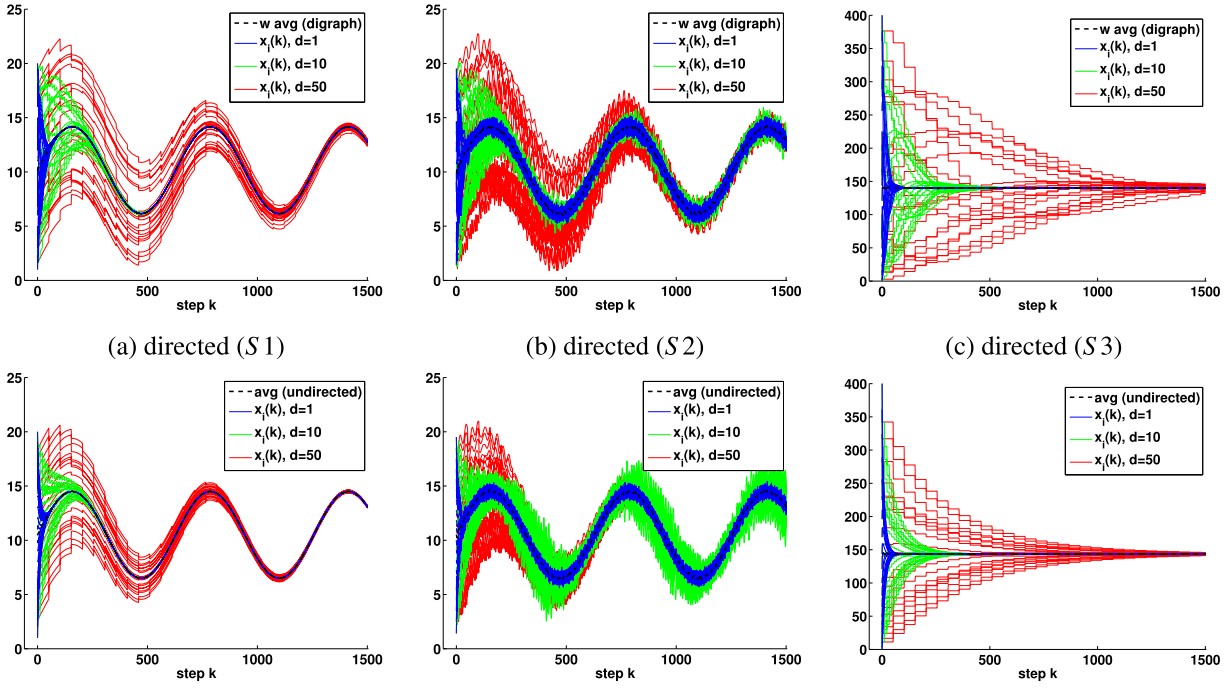
- ($S2$) Time-varying reference signals with different time-variation pattern.

$$r_i(k) = i + 4\sin(0.01k) + 0.5\sin\left(\frac{k}{\sqrt{i}} + \sqrt{i}\right), \quad \forall i = 1, \dots, n, \ \forall k \geq 0. \tag{73}$$



(a) References ($S1$) (b) References ($S2$) (c) References ($S3$)

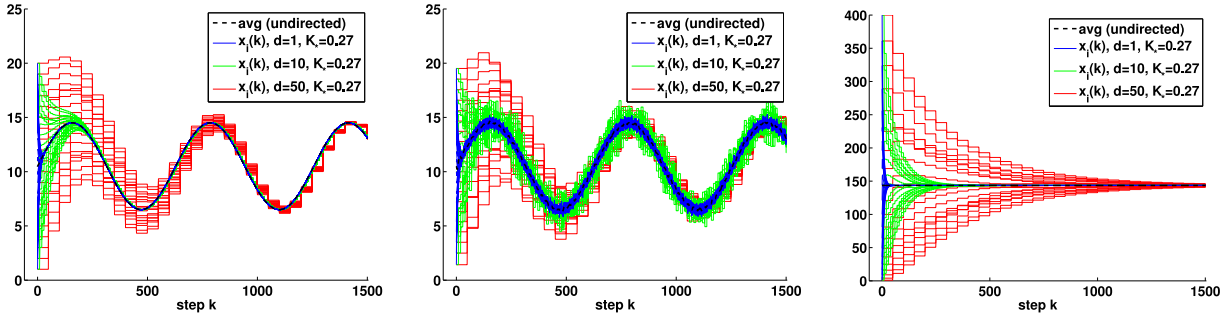
Fig. 4. References measured by the robots (gray solid), along the iterations k (abscissa–axis). The weighted average $r_{wavg}(k)$ and average $r_{avg}(k)$ of the reference signals given by (2) and (3) are shown respectively in red solid and in black solid. These average values are associated with, respectively, the directed strongly connected and the undirected connected graphs in Fig. 3.



(a) directed ($S1$) (b) directed ($S2$) (c) directed ($S3$)

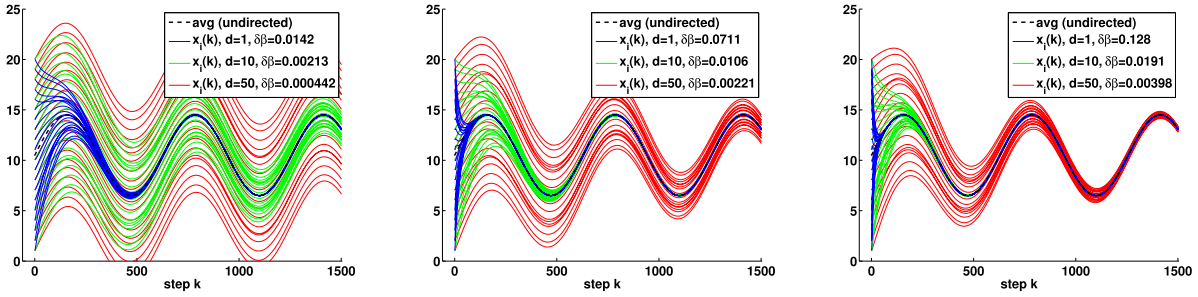
(d) undirected ($S1$) (e) undirected ($S2$) (f) undirected ($S3$)

Fig. 5. Proposed dynamic consensus with delay compensation for the cases ($S1$, $S2$, $S3$) in Fig. 4, under the directed ((a),(b),(c)) and the undirected communication graphs ((d), (e), (f)) in Fig. 3. The states of each agent $i = 1, \dots, n$ along the iterations k (abscissa–axis), for different delay values $d \in \{1, 10, 50\}$ are shown in different colors. The weighted average $r_{wavg}(k)$ (directed graph: (a), (b), (c)) and the average $r_{avg}(k)$ (undirected graph: (d), (e), (f)), of the time–varying reference signals in (2) and (3) are displayed in black dashed.



(a) $S1$ (b) $S2$ (c) $S3$

Fig. 6. Alternative execution. Instead of using the proposed dynamic consensus with delay compensation, agents use here a dynamic consensus method without delay compensation [1,2,30], (7). Here, as discussed in Section 2.2, agents use a larger period for trying to get rid of the delay d, for the cases ($S1$, $S2$, $S3$) in Fig. 4 under the undirected communication graph (Fig. 3(b)). The states of each agent $i = 1, \dots, n$, along the iterations k (abscissa–axis), are here depicted for different delay values $d \in \{1, 10, 50\}$ in different colors.



(a) $\beta\delta = 0.1(2/\lambda_{\mathcal{L}_n})\sin(\pi/(2(2d+1)))$ (b) $\beta\delta = 0.5(2/\lambda_{\mathcal{L}_n})\sin(\pi/(2(2d+1)))$ (c) $\beta\delta = 0.9(2/\lambda_{\mathcal{L}_n})\sin(\pi/(2(2d+1)))$

Fig. 7. Alternative execution. Instead of using the proposed dynamic consensus with delay compensation, agents use here a dynamic consensus method without delay compensation [1,2,30], (7), for the simulated case ($S1$) in Fig. 4(a) under the undirected communication graph (Fig. 3(b)). The states of each agent $i = 1, \dots, n$ (solid, different colors), along the iterations k (abscissa–axis), are here depicted, together with the average of the time-varying reference signals (black dashed).

- ($S3$) Time–constant reference signals with different values.

$$r_i(k) = i^2,\ \forall i = 1, \dots, n,\ \forall k \geq 0. \tag{74}$$

In the first set of simulations ($S1$), the references measured by every robot $i = 1, \dots, n$ are time–varying. Note however the definition of α_∞^+ in (64). Here, for all agent i and every step k, the term

$$r_i(k) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^{n} [w_{\mathcal{L}_1}]_j r_j(k) = i - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^{n} [w_{\mathcal{L}_1}]_j j,$$

so that

$$\left(r_i(k) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^{n} [w_{\mathcal{L}_1}]_j r_j(k) \right)$$
$$- \left(r_i(k-1-d) - \frac{1}{\mathbf{1}^T w_{\mathcal{L}_1}} \sum_{j=1}^{n} [w_{\mathcal{L}_1}]_j r_j(k-1-d) \right) = 0,$$

and $\alpha_\infty^+ = 0$. Thus, if the system matrix is stable, then robots will asymptotically track $r_{wavg}(k)$ in (2) with zero error (the ultimate tracking bound $\mathcal{B}$ in (53) is zero). In the second set of simulations ($S2$), the references are again time–varying. The value of α_∞^+ in (64) is different from zero, i.e., $\alpha_\infty^+ > 0$, so that if the system matrix is stable, then robots will asymptotically track $r_{wavg}(k)$ in (2) with an ultimate tracking error value $\mathcal{B}$ (53). In the third set of simulations ($S3$), the references remain constant over time, similar as for a static consensus situation. The ultimate tracking error $\mathcal{B}$ is zero, since $\alpha_\infty^+ = 0$ in (64).

Fig. 4(a) to (c) shows the reference signals (solid gray) for each case, the weighted average $r_{wavg}(k)$ (solid red), and the average $r_{avg}(k)$ (solid black) of the reference signals given by (2) and (3), for the cases in which the communication graph is, respectively, directed strongly connected (Fig. 3(a)) and undirected connected (Fig. 3(b)). Recall that, for undirected graphs, $\mathbf{1}^T$ is a left eigenvector of the Laplacian $\mathcal{L}$ associated with the eigenvalue 0, i.e., $\mathbf{1}^T\mathcal{L} = \mathbf{0}^T$. Thus, (2) is the exact average (3).

Fig. 5(a), (b), (c) shows the results of the proposed dynamic consensus with delay compensation (10) for the directed communication graph in Fig. 3(a), for the three scenarios in Fig. 4, and for different delays $d \in \{1, 10, 50\}$. For the digraph (Fig. 3(a)), the associated Laplacian has eigenvalues with the following characteristics (Definition 2): $\lambda_{\mathcal{L}_{\cos}} = 0.7497$, $\lambda_{\mathcal{L}_{\min}} = 0.3869$ (associated with $\lambda_{\mathcal{L}_i} = 0.29 - 0.256i$), and $\lambda_{\mathcal{L}_{\max}} = 3.4197$ (associated with $\lambda_{\mathcal{L}_i} = 3.4197$). We set the parameters of the proposed dynamic average consensus with delay compensation algorithm (10), (25) as follows: K and γ according to Theorem 1, i.e., $K = K_\star = \frac{2\lambda_{\mathcal{L}_{\cos}}}{(\lambda_{\mathcal{L}_{\min}}+\lambda_{\mathcal{L}_{\max}})} = 0.3939$, $\gamma_\star = 1/K_\star = 2.5387$. We show the evolution of the states of each robot $i \in \{1, \dots, n\}$ for the different delays d in different colors. As stated by Theorem 1, in all the scenarios and for all the delay values d, the states converge to a neighborhood of the weighted average $r_{wavg}(k)$ in (2) (black dashed). In addition, in scenarios $S1$ and $S3$, since α_∞^+ equals zero as previously commented, the states of the robots asymptotically converge exactly to $r_{wavg}(k)$. Observe also that, as stated by Theorem 1, the convergence is faster for lower delay values d. As the delay increases, the convergence speed is reduced, but the system remains stable. For each delay, we obtain the following theoretical upper bound for the convergence speed (Theorem 1, (63)),

$$\rho(M_1) \leq \left(1 - \frac{4(\lambda_{\mathcal{L}_{\cos}})^2 \lambda_{\mathcal{L}_{\min}} \lambda_{\mathcal{L}_{\max}}}{(\lambda_{\mathcal{L}_{\min}} + \lambda_{\mathcal{L}_{\max}})^2} \right)^{\frac{1}{2(d+1)}} = 0.7947^{\frac{1}{2(d+1)}},$$

and compare it against the true $\rho(M_1)$ obtained for the augmented system matrix M_1:

$$\rho(M_1) = 0.9442 \leq 0.7947^{\frac{1}{2(d+1)}}|_{d=1} = 0.9442,$$
$$\rho(M_1) = 0.9896 \leq 0.7947^{\frac{1}{2(d+1)}}|_{d=10} = 0.9896,$$
$$\rho(M_1) = 0.9977 \leq 0.7947^{\frac{1}{2(d+1)}}|_{d=50} = 0.9977.$$

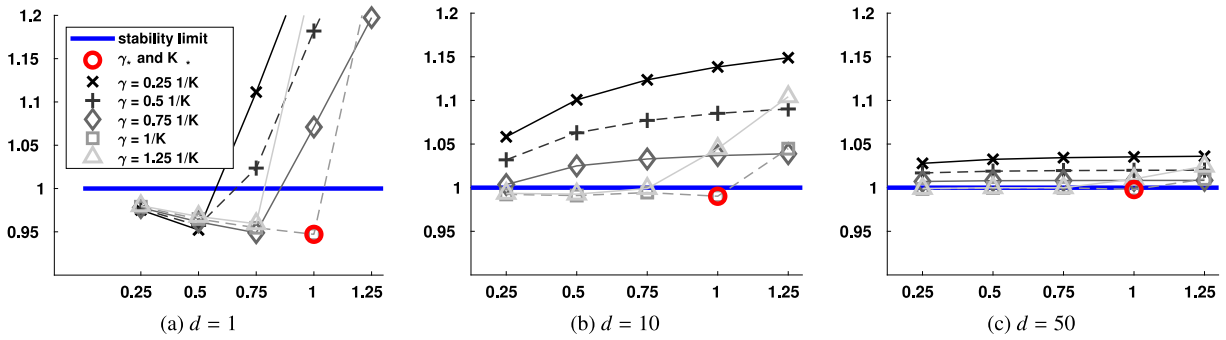
In this particular case, both numerical and theoretical values are exactly equal, since the $\lambda_{\mathcal{L}_{\cos}}$ value was obtained for the same eigenvalue that produced $\lambda_{\mathcal{L}_{\min}}$.

Fig. 5(d), (e), (f) shows the results of the proposed dynamic consensus with delay compensation (10) for the undirected communication graph in Fig. 3(b), for the three scenarios in Fig. 4, and for different delays $d \in \{1, 10, 50\}$. For the undirected graph (Fig. 3(b)), the associated Laplacian has the eigenvalues $\lambda_{\mathcal{L}_2} = 0.3820$ and $\lambda_{\mathcal{L}_n} = 7.0322$. We set the parameters of the proposed dynamic average consensus with delay compensation algorithm (10), (25) as follows: K and γ according to Corollary 1, i.e., $K = K_\star = \frac{2}{(\lambda_{\mathcal{L}_2}+\lambda_{\mathcal{L}_n})} = 0.2698$, $\gamma_\star = 1/K_\star = 3.7064$. We show the evolution of the states of each robot $i \in \{1, \dots, n\}$ for the different delays d in different colors. As stated by Theorem 1, in all the scenarios and for all the delay values d, the states converge to a neighborhood of the average $r_{avg}(k)$ in (3) (black dashed). In addition, in scenarios $S1$ and $S3$, since α_∞^+ equals zero as previously commented, the states of the robots asymptotically converge exactly to $r_{avg}(k)$ in (3). Observe also that, as stated by Corollary 1, the convergence is faster for lower delay values d. As the delay increases, the convergence speed is reduced, but the system remains stable. For each delay, we obtain the following theoretical convergence speed (Corollary 1, (71)),

$$\rho(M_1) = \left(\frac{\lambda_{\mathcal{L}_n} - \lambda_{\mathcal{L}_2}}{\lambda_{\mathcal{L}_n} + \lambda_{\mathcal{L}_2}} \right)^{\left(\frac{1}{d+1}\right)} = 0.897^{\left(\frac{1}{d+1}\right)},$$

and compare it against the true $\rho(M_1)$ obtained for the augmented system matrix M_1:

$$\rho(M_1) = 0.9471 = 0.897^{\left(\frac{1}{d+1}\right)}|_{d=1} = 0.9471,$$
$$\rho(M_1) = 0.9902 = 0.897^{\left(\frac{1}{d+1}\right)}|_{d=10} = 0.9902,$$
$$\rho(M_1) = 0.997 = 0.897^{\left(\frac{1}{d+1}\right)}|_{d=50} = 0.9979,$$



(a) $d = 1$ (b) $d = 10$ (c) $d = 50$

Fig. 8. Convergence speed when the delay equals $d = 1$ (a), $d = 10$ (b) and $d = 50$ (c) for the communication graph in Fig. 3(b), which is undirected. The figures contain the graphical representation of the values shown in Table 1. They represent the modulus of the leading eigenvalue. Values closer to 0 provide faster convergence speed. Values larger than one (stability limit, in blue), are associated with unstable systems. The abscissa–axis represents different values of K, which are equal to $K = factor_K K_\star$, where $factor_K \in \{0.25, 0.5, 0.75, 1, 1.25\}$. Every plot represents a value of γ equal to $\gamma = \frac{factor_\gamma}{K}$, where $factor_\gamma \in \{0.25, 0.5, 0.75, 1, 1.25\}$, as shown in the legend. We show in red the convergence speed obtained with the proposed parameter tuning ($K_\star$, $\gamma_\star$ in Corollary 1), which is the smallest one, and thus, produces the fastest speed. Note also that it is always stable (smaller than one, blue line).

obtaining in this case the same values for the numerical and theoretical values, as expected from Corollary 1.

All the simulations in this section were run in a computer with Matlab 2012b, Windows 10, processor 11th Gen Intel(R) Core(TM) i5-1135G7 2.40 GHz, 8,00 GB RAM. Next, we include an example of the execution times needed to run the sets of simulations displayed in Fig. 5(a), which are similar to the ones obtained in the other cases (b) to (f). These times are associated with $n = 20$ agents, running 1500 iterations of the method. For $d = 1$, $d = 10$ and $d = 50$, the execution times were respectively 1.394 s, 2.419 s and 6.814 s. Thus, for $d = 50$, the execution time per agent and iteration is in average 0.227 ms (6.814 s/1500 iterations/20 agents). This confirms that, as commented in Remark 2, the proposed algorithm is light in terms of the required communication, computation, and memory resources.

6.1. Effects of parameter tuning on stability

Now, we make a study of the different speeds of convergence that can be achieved with our proposed dynamic average consensus with delay compensation method (10), (25), if γ and K are selected with values different from the ones in Theorem 1 and Corollary 1 ($\gamma_\star$ and $K_\star$). We consider the communication graph in Fig. 3(b). In Table 1 and Fig. 8, we show the leading eigenvalues of the augmented system matrix M, excluding the ones associated with the Laplacian eigenvalue $\lambda_{\mathcal{L}_1} = 0$, for delays $d \in \{1, 10, 50\}$. Recall that the proposed method is oriented towards compensating the negative effects of the delays, so that, under mild conditions on the connectivity of the graphs and on the reference signals, we ensure convergence and improve the convergence speed. Both ideas (stability and convergence speed) are properly represented by $\rho(M_1)$ in Eqs. (63) and (71). As discussed in Proposition 3 and its proof, the stability and convergence speed of the algorithm is related to the speed with which $\rho(M_1)^k$ decays to zero. Thus, if $\rho(M_1) < 1$, then $\rho(M_1)^k$ asymptotically becomes zero (otherwise, it grows up as k increases). The convergence is faster when $\rho(M_1)$ is closer to zero. For this reason, we select this value as the performance evaluation indicator in this section.

Observe that the resulting system matrix is unstable in several cases (values larger than one in Table 1, values above the blue line in Fig. 8). The behavior of the eigenvalues is counter-intuitive unless they are analyzed in terms, e.g., of the root locus, as done in Section 5. As stated in Theorem 1 and Corollary 1, the selection $\gamma_\star$ and $K_\star$ (in bold) always give rise to a stable system matrix. The improvement of the convergence speed was justified for undirected graphs and large values of d in Section 5. We observe this in Table 1 (center, $d = 10$) and (bottom, $d = 50$), where the achieved speed using $\gamma_\star$ and $K_\star$ (in bold) is faster than for the other selections. Observe also in Fig. 8 that the smallest eigenvalue modulus and thus, the fastest convergence speed, is obtained when using $\gamma_\star$ and $K_\star$ from Corollary 1 (red circle). In this example, faster convergence speed can also be appreciated for small delays (Table 1 (top, $d = 1$)).

Table 1
Convergence speed when the delay equals $d = 1$ (top), $d = 10$ (center), and $d = 50$ (bottom), for different values of K and γ. Each column represents a value of K equal to $K = factor_K K_\star$, where $factor_K \in \{0.25, 0.5, 0.75, 1, 1.25\}$ is the number shown in the top row of the table. Each row represents a value of γ equal to $\gamma = \frac{factor_\gamma}{K}$, where $factor_\gamma \in \{0.25, 0.5, 0.75, 1, 1.25\}$ is the number shown in the left column of the table.

$d = 1$					
γ \ K	**0.25** $K_\star$	**0.5** $K_\star$	**0.75** $K_\star$	**1** $K_\star$	**1.25** $K_\star$
0.25 $\frac{1}{K}$	0.9752	0.9522	1.1114	1.2833	1.4348
0.5 $\frac{1}{K}$	0.9766	0.9573	1.0236	1.1819	1.3214
0.75 $\frac{1}{K}$	0.9779	0.9615	0.9489	1.0710	1.1974
1 $\frac{1}{K}$	0.9790	0.9648	0.9547	**0.9471**	1.250
1.25 $\frac{1}{K}$	0.9801	0.9677	0.9593	1.250	1.5625
$d = 10$					
γ \ K	**0.25** $K_\star$	**0.5** $K_\star$	**0.75** $K_\star$	**1** $K_\star$	**1.25** $K_\star$
0.25 $\frac{1}{K}$	1.0583	1.1007	1.1236	1.1384	1.1489
0.5 $\frac{1}{K}$	1.0319	1.0629	1.0771	1.0852	1.0902
0.75 $\frac{1}{K}$	1.0036	1.0250	1.0329	1.0368	1.0389
1 $\frac{1}{K}$	0.9923	0.9910	0.9944	**0.9902**	1.0447
1.25 $\frac{1}{K}$	0.9935	0.9926	0.9990	1.0447	1.1042
$d = 50$					
γ \ K	**0.25** $K_\star$	**0.5** $K_\star$	**0.75** $K_\star$	**1** $K_\star$	**1.25** $K_\star$
0.25 $\frac{1}{K}$	1.0279	1.0325	1.0344	1.0353	1.0360
0.5 $\frac{1}{K}$	1.0169	1.0189	1.0196	1.0199	1.0201
0.75 $\frac{1}{K}$	1.0071	1.0080	1.0082	1.0083	1.0084
1 $\frac{1}{K}$	0.9986	0.9997	0.9999	**0.9979**	1.0100
1.25 $\frac{1}{K}$	0.9991	0.9999	0.9999	1.0100	1.0246

6.2. Effects of sparseness on parameter tuning

We have performed several simulations to demonstrate the performance of the proposed method for different graph topologies, including undirected graphs that may be sparsely connected. As shown in Fig. 9(a), (b), (c), we have performed simulations with a Minimum distance Spanning Tree (Graph 1, in red). We have also used Graph 2, in green, which includes all the links in Graph 1, and also some additional

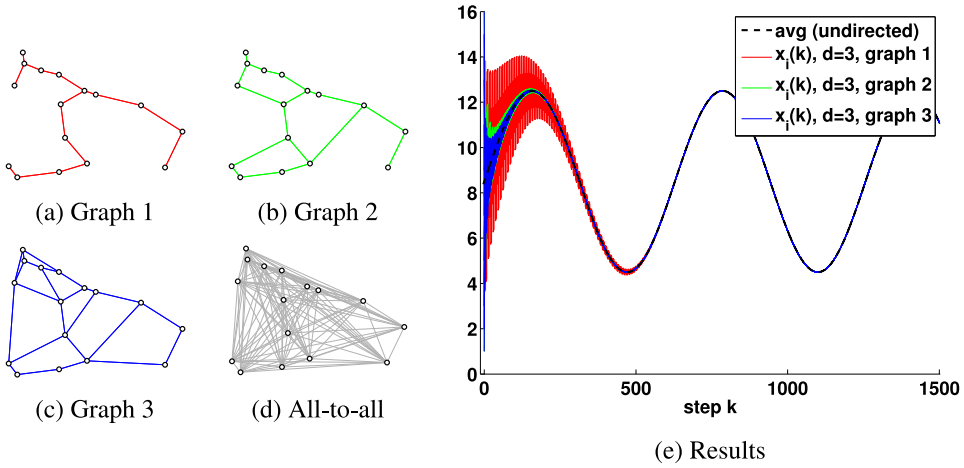


Fig. 9. Sparse communication graphs. (a) Graph 1 is a Minimum-distance Spanning Tree (MST). (b) Graph 2 contains all the edges in Graph 1, and some additional links. (c) Graph 3 is more connected than Graph 2. (d) Graph with all-to-all connections. (e) Results of the proposed consensus with delay compensation for the sparse graphs in Fig. 9(a), (b), (c). The states of each agent $i = 1, \dots, n$ along the iterations k (abscissa–axis), for the different graphs, are shown in different colors. The average $r_{avg}(k)$, of the time-varying reference signals in (3) are displayed in black dashed.

links. We have used a third graph (Graph 3) even more connected (in blue). Note that all these graphs are connected and that all of them are sparse. Consider instead the graph with all-to-all connections which appears in Fig. 9(d), in gray.

We have executed simulations using $d = 3$ delay steps, to track the average of the reference signals in scenario $S1$ (72), using the three sparse graph topologies in Fig. 9(a), (b), (c). Since the three sparse graphs are undirected, we use Corollary 1 to tune the parameters $K = K_\star$ and $\gamma = \gamma_\star$. Using Graph 1 (Fig. 9(a)), which is a MST tree graph, we get the following eigenvalues associated with the graph Laplacian: $\lambda_{\mathcal{L}_2} = 0.0679$ and $\lambda_{\mathcal{L}_n} = 4.5235$. The parameter tuning values obtained, following Corollary 1, are $K = K_\star = 0.4356$, $\gamma = \gamma_\star = 0.1435$. We obtain the following convergence speed: $\rho(M_1) = 0.9925$. Fig. 9(e), in red, shows the evolution of the states of the nodes along the steps. Note that the states converge asymptotically to the average of the reference values (in black dashed), regardless the presence of the delay $d = 3$.

We use now Graph 2 (Fig. 9(b)), that is more connected than Graph 1. The Laplacian eigenvalues and the result of the parameter tuning using Corollary 1 are: $\lambda_{\mathcal{L}_2} = 0.1819$ and $\lambda_{\mathcal{L}_n} = 5.2177$, $K = K_\star = 0.3704$, $\gamma = \gamma_\star = 0.1687$. For this topology, we obtain the following convergence speed $\rho(M_1) = 0.9827$. Observe that this speed is faster than for Graph 1, since $\rho(M_1)$ has a value closer to zero. Since this graph has more links, in the absence of delays, the convergence of the consensus algorithms would be faster. As it can be observed, since the parameter tuning proposed in this paper is oriented towards improving the convergence speed counteracting the negative effects of the delays d, it also gives rise to faster convergence in the presence of delays when the graphs have higher connectivity. Fig. 9(e), in green, shows the evolution of the states of the nodes along the steps. The states (in green) converge asymptotically to the average of the reference values (in black dashed), with a faster speed than for Graph 1 (in red).

Finally, we use Graph 3 (Fig. 9(c)), which is the sparse graph with the highest connectivity in this set of simulations. The Laplacian eigenvalues and the result of the parameter tuning using Corollary 1 are $\lambda_{\mathcal{L}_2} = 0.3495$ and $\lambda_{\mathcal{L}_n} = 6.6138$, $K = K_\star = 0.2872$, $\gamma = \gamma_\star = 0.2176$. For this topology, we obtain the fastest convergence speed rate $\rho(M_1) = 0.9739$. We can see that the states of the nodes (Fig. 9(e), in blue) converge asymptotically to the average of the reference values (in black dashed), and faster than for Graphs 1 (in red) and 2 (in green).

6.3. Comparison with other alternatives

We compare the performance of the proposed algorithm with other alternatives. Fig. 6 shows an alternative execution, commented in Section 2.2 in which agents do not run the proposed dynamic consensus with delay compensation. Instead, they run a dynamic consensus method without delay compensation [1,2,30], (7). Besides, as explained in Section 2.2, instead of using a measurement and update period T associated with each iteration k, a larger period dT is used. This means that the references are sampled less often. On the other hand, nodes wait until they receive (or process) the relative states from neighbors so that now the update refers to consecutive steps $k', k'-1$ (i.e., they get rid of the delay d). In this example, we consider the cases ($S1$, $S2$, $S3$) in Fig. 4 under the undirected communication graph (Fig. 3(b)). For the case $S3$ (static consensus) there is no difference. However, observe for cases $S1$ and $S2$ (dynamic consensus) the negative effects on the robot states due to the decreased sampling frequency. The robot states remain unchanged for a longer time (here, dT, compared to Fig. 5((d), (e), (f)), in which the states change at every step; here, we take $T = 1$). As a result, the average is tracked with less precision. This problem does not appear in static consensus, but it is important in dynamic consensus scenarios. Note also that the benefits of the proposed method (Fig. 5(d), (e), (f)) relative to Fig. 6 become more important for larger delay values d.

In Fig. 7, we compare the performance of the proposed algorithm with dynamic consensus methods that do not include delay compensation [1,2,30], (7), for the simulated case ($S1$) in Fig. 4(a) under the undirected communication graph (Fig. 3(b)). In all cases, we use values of $\beta\delta$ that satisfy (9). Note that these values depend on the delay d, and that it is unclear from (9) which value will produce faster convergence. The proposed method (Fig. 5(d)) has a single value $K_\star$ for all the delay values d and converges faster.

Finally, Fig. 10 shows another case in which agents do not use the proposed dynamic consensus with delay compensation method. Instead, they consider (7) but without any delays in the state of agent i, i.e., making $u_i(k) = r_i(k) - r_i(k-1) - \beta\delta \sum_{j=1}^{n} a_{ij}(x_i(k-1) - x_j(k-1-d))$. Static consensus methods in which the self-information from agents i is not delayed, have been reported to be convergent regardless of the delay d. We show an immediate application to the dynamic consensus protocol in (7) of this idea. In all the simulations, we used $\beta\delta = 0.5/\mathcal{N}_{\max}$ that satisfies (5) (for the undirected graph in Fig. 3(b), the maximum node degree is $\mathcal{N}_{\max} = 4$). Whereas for all the tested delays $d \in \{1, 10, 50\}$ the resulting system converged, it can also be seen that the states of the agents no longer track the average of the reference signals. Thus, this option should be discarded, unless a careful tuning was carried out to ensure the states track indeed the average.

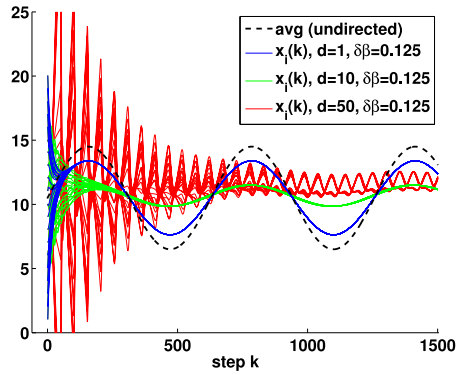


Fig. 10. Dynamic consensus without delay compensation for the simulated case ($S1$) in Fig. 4(a) under the undirected communication graph (Fig. 3(b)). In these simulations, the delay d only affects the neighbors' states, but not the local state of agents. The states of each agent $i = 1, \dots, n$ (solid, different colors), along the iterations k (abscissa–axis), are here depicted, together with the average of the time–varying reference signals (black dashed).

7. Conclusions

We have presented a dynamic consensus method with delay compensation. The design of the consensus parameters has also been discussed to ensure the convergence of the closed–loop systems for arbitrarily large fixed delays d. This is a strong benefit since dynamic consensus methods without delay compensation diverge for large delays. Remarkably, the tuning criteria of the consensus parameters do not depend on the delay d. We have also given a characterization of the convergence speed of our method. Simulation results have been provided to show the closed–loop performance for systems with $n = 20$ agents and large delays. The proposed algorithm depends strongly on the fact that all delays have the same time–constant value d and are known in advance (see Assumption 1) by the agents. We have discussed examples of applications in which the proposed method is of interest. In order to cope with a wider variety of scenarios, future extensions of this work might consider time–varying delays [5]. In addition, the graphs considered here include a wide variety of topologies (see Assumption 2), such as strongly connected directed graphs. In fact, we have performed several simulations to demonstrate the performance of the proposed method for different graph topologies, including undirected graphs that may be sparsely connected. An interesting future extension would consider time–varying topologies, which are quite challenging in the context of dynamic average consensus methods.

CRediT authorship contribution statement

Rosario Aragues: Conceptualization, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Antonio González:** Conceptualization, Formal analysis, Investigation, Writing – original draft. **Gonzalo López–Nicolás:** Methodology, Validation, Writing – review & editing, Project administration. **Carlos Sagues:** Methodology, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rosario Aragues reports financial support was provided by Spain Ministry of Science and Innovation. Antonio Gonzalez, Gonzalo Lopez-Nicolas, Carlos Sagues reports financial support was provided by Spain Ministry of Science and Innovation. Rosario Aragues reports financial support was provided by Government of Aragon Department of Science Technology and University. Antonio Gonzalez, Gonzalo Lopez-Nicolas, Carlos Sagues reports financial support was provided by Government of Aragon Department of Science Technology and University. Rosario Aragues reports financial support was provided by European Union. Antonio Gonzalez, Gonzalo Lopez-Nicolas, Carlos Sagues reports financial support was provided by European Union.

Data availability

No data was used for the research described in the article.

Acknowledgments

Supported via group Gobierno de Aragón, Spain T45_23R and projects PID2021-124137OB-I00, PID2020-116585GB-I00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR.

References

[1] S.S. Kia, B. Van Scoy, J. Cortes, R.A. Freeman, K.M. Lynch, S. Martinez, Tutorial on dynamic average consensus: The problem, its applications, and the algorithms, IEEE Control Syst. Mag. 39 (3) (2019) 40–72.
[2] S.S. Kia, J. Cortés, S. Martinez, Dynamic average consensus under limited control authority and privacy requirements, Internat. J. Robust Nonlinear Control 25 (13) (2015) 1941–1966.
[3] R.A. Freeman, P. Yang, K.M. Lynch, Stability and convergence properties of dynamic average consensus estimators, in: IEEE Conf. on Decision and Control, 2006, pp. 338–343.
[4] L. Xiao, S. Boyd, S. Lall, A space-time diffusion scheme for peer-to-peer least-squares estimation, in: Int. Conf. on Information Processing in Sensor Networks, 2006, pp. 168–176.
[5] A. González, R. Aragues, G. López-Nicolás, C. Sagüés, Predictor-feedback synthesis in coordinate-free formation control under time-varying delays, Automatica 113 (2020) 108811.
[6] M. Todescato, N. Bof, G. Cavraro, R. Carli, L. Schenato, Partition-based multi-agent optimization in the presence of lossy and asynchronous communication, Automatica 111 (2020) 108648.
[7] L. Ji, L. Yu, C. Zhang, X. Guo, H. Li, Initialization-free distributed prescribed-time consensus based algorithm for economic dispatch problem over directed network, Neurocomputing 533 (2023) 1–9.
[8] S. Chen, D.W. Ho, Information-based distributed extended Kalman filter with dynamic quantization via communication channels, Neurocomputing 469 (2022) 251–260.
[9] W. Wang, Mean-square exponential input-to-state stability of stochastic fuzzy delayed Cohen-Grossberg neural networks, J. Exp. Theor. Artif. Intell. (2023) 1–14.
[10] C. Huang, B. Liu, C. Qian, J. Cao, Stability on positive pseudo almost periodic solutions of HPDCNNs incorporating D operator, Math. Comput. Simulation 190 (2021) 1150–1163.
[11] B. Liu, Global exponential stability for BAM neural networks with time-varying delays in the leakage terms, Nonlinear Anal. RWA 14 (1) (2013) 559–566.
[12] Q. Zhu, J. Cao, Mean-square exponential input-to-state stability of stochastic delayed neural networks, Neurocomputing 131 (2014) 157–163.
[13] C. Zhou, H. Tao, Y. Chen, V. Stojanovic, W. Paszke, Robust point-to-point iterative learning control for constrained systems: A minimum energy approach, Internat. J. Robust Nonlinear Control 32 (18) (2022) 10139–10161.
[14] Z. Zhuang, H. Tao, Y. Chen, V. Stojanovic, W. Paszke, An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints, IEEE Trans. Syst. Man Cybern. A 53 (6) (2023) 3461–3473.
[15] T. Wei, X. Li, V. Stojanovic, Input-to-state stability of impulsive reaction–diffusion neural networks with infinite distributed delays, Nonlinear Dynam. 103 (2021) 1733–1755.
[16] Y. Ghaedsharaf, M. Siami, C. Somarakis, N. Motee, Centrality in time-delay consensus networks with structured uncertainties, Automatica 125 (2021) 109378.
[17] F. Chen, C. Chen, G. Guo, C. Hua, G. Chen, Delay and packet-drop tolerant multistage distributed average tracking in mean square, IEEE Trans. Cybern. 52 (9) (2021) 9535–9545.
[18] S. Jafarizadeh, D. Veitch, Robust weighted-average continuous-time consensus with communication time delay, IEEE Trans. Cybern. (2021).
[19] S. De, S.R. Sahoo, P. Wahi, Communication-delay-dependent rendezvous with possible negative controller gain in cyclic pursuit, IEEE Trans. Control Netw. Syst. 7 (3) (2019) 1069–1079.

[20] G. Wen, Y. Yu, Z. Peng, H. Wang, Dynamical group consensus of heterogenous multi-agent systems with input time delays, Neurocomputing 175 (2016) 278–286.
[21] Z. Wang, J. Xu, X. Song, H. Zhang, Consensus problem in multi-agent systems under delayed information, Neurocomputing 316 (2018) 277–283.
[22] J. Xu, H. Zhang, L. Xie, Input delay margin for consensusability of multi-agent systems, Automatica 49 (6) (2013) 1816–1820.
[23] Y.G. Sun, L. Wang, G. Xie, Average consensus in networks of dynamic agents with switching topologies and multiple time-varying delays, Systems Control Lett. 57 (2) (2008) 175–183.
[24] Y.-P. Tian, C.-L. Liu, Consensus of multi-agent systems with diverse input and communication delays, IEEE Trans. Automat. Control 53 (9) (2008) 2122–2128.
[25] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (1) (2007) 215–233.
[26] Z. Wang, J. Xu, X. Song, H. Zhang, Consensus conditions for multi-agent systems under delayed information, IEEE Trans. Circuits Syst. II 65 (11) (2018) 1773–1777.
[27] W. Qiao, R. Sipahi, Delay-dependent coupling for a multi-agent LTI consensus system with inter-agent delays, Physica D 267 (2014) 112–122.
[28] P.-A. Bliman, G. Ferrari-Trecate, Average consensus problems in networks of agents with delayed communications, Automatica 44 (8) (2008) 1985–1995.
[29] Y. Ghaedsharaf, N. Motee, Performance improvement in time-delay linear consensus networks, in: American Control Conference, 2017, pp. 2345–2350.
[30] H. Moradian, S.S. Kia, On robustness analysis of a dynamic average consensus algorithm to communication delay, IEEE Trans. Control Netw. Syst. 6 (2) (2019) 633–641.
[31] A. Nedić, A. Ozdaglar, Convergence rate for consensus with delays, J. Global Optim. 47 (3) (2010) 437–456.
[32] L. Moreau, Stability of continuous-time distributed consensus algorithms, in: IEEE Conf. on Decision and Control, 2004, pp. 3998–4003.
[33] A. Seuret, D.V. Dimarogonas, K.H. Johansson, Consensus under communication delays, in: IEEE Conf. on Decision and Control, 2008, pp. 4922–4927.
[34] T. Charalambous, C.N. Hadjicostis, Average consensus in the presence of dynamically changing directed topologies and time delays, in: IEEE Conf. on Decision and Control, 2014, pp. 709–714.
[35] C. Somarakis, J.S. Baras, Delay-independent stability of consensus networks with application to flocking, IFAC-PapersOnLine 48 (12) (2015) 159–164.
[36] A. González, M. Aranda, G. López-Nicolás, C. Sagüés, Time delay compensation based on smith predictor in multiagent formation control, IFAC-PapersOnLine 50 (1) (2017) 11645–11651.
[37] A. Gonzalez, P. Garcia, P. Albertos, P. Castillo, R. Lozano, Robustness of a discrete-time predictor-based controller for time-varying measurement delay, Control Eng. Pract. 20 (2) (2012) 102–110.
[38] A. González, A. Sala, R. Sanchis, LK stability analysis of predictor-based controllers for discrete-time systems with time-varying actuator delay, Systems Control Lett. 62 (9) (2013) 764–769.
[39] Z. Artstein, Linear systems with delayed controls: A reduction, IEEE Trans. Automat. Control 27 (4) (1982) 869–879.
[40] Z. Wang, H. Zhang, X. Song, H. Zhang, Consensus problems for discrete-time agents with communication delay, Int. J. Control Autom. Syst. 15 (4) (2017) 1515–1523.
[41] B. Zhou, Z. Lin, Consensus of high-order multi-agent systems with large input and communication delays, Automatica 50 (2) (2014) 452–464.
[42] A. Ponomarev, Z. Chen, H.-T. Zhang, Discrete-time predictor feedback for consensus of multiagent systems with delays, IEEE Trans. Automat. Control 63 (2) (2018) 498–504.
[43] S. Chen, Z. Zou, Z. Zhang, L. Zhao, Fixed-time scaled consensus of multi-agent systems with input delay, J. Franklin Inst. B (2022).
[44] J.B. Ernst, Energy-efficient next-generation wireless communications, Handb. Green Inf. Commun. Syst. (2012) 371.
[45] T.H. Illangasekare, Q. Han, A.P. Jayasumana, Environmental underground sensing and monitoring, in: Underground Sensing, Elsevier, 2018, pp. 203–246.
[46] T.M.D. Tran, A.Y. Kibangou, Distributed estimation of Laplacian eigenvalues via constrained consensus optimization problems, Systems Control Lett. 80 (2015) 56–62.
[47] A. Gusrialdi, Z. Qu, Distributed estimation of all the eigenvalues and eigenvectors of matrices associated with strongly connected digraphs, IEEE Control Syst. Lett. 1 (2) (2017) 328–333.
[48] Z. Lin, On asymptotic stabilizability of discrete-time linear systems with delayed input, in: IEEE Int. Conf. on Control and Automation, 2007, pp. 432–437.
[49] D. Seborg, T. Edgar, D. Mellichamp, Process Dynamics and Control, first ed., Chapter 26: Design of Digital Controllers, Wiley, 1989.

Rosario Aragues received the M.S. and Ph.D. degrees in System Engineering and Computer Science from Univ. Zaragoza, Spain, in 2008 and 2012. She was a postdoctoral researcher at the Institut Pascal, CNRS, UMR 6602, Clermont-Ferrand, France, from April 2012 to October 2013. Currently she works as an Associate Professor with the Department of Computer Science and Systems Engineering at the University of Zaragoza. Her research interests include multi–robot perception and control using distributed and decentralized strategies.

Antonio González received the degree (Hons.) in Telecommunications Engineering and the Ph.D. degree in Automation and Industrial Informatics from the Universitat Politècnica de València (UPV), Spain, in 2012. He was a Postdoctoral Researcher with the Laboratory of Industrial and Human Automation control, Mechanical Engineering and Computer Science, CNRS, UMR 8201, Valenciennes, France, from 2013 to 2014. He currently works as an Associate Professor with the Department of Systems Engineering and Automation, Universitat Politècnica de València. His research interests include the broad area of time delay systems, robust control, networked control systems, multi–robot systems, and process control applications.

Gonzalo López-Nicolás received the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Zaragoza, Spain, in 2008. He is currently a Full Professor with the Department of Computer Science and Systems Engineering, University of Zaragoza. He is a member of the Robotics, Perception, and Real-Time Group, and the Aragon Institute of Engineering Research (I3A). His current research interests include visual control, autonomous robot navigation, multi-robot systems, and the application of computer vision techniques to robotics.

Carlos Sagues received the M.Sc. degree in computer science and systems engineering and the Ph.D. degree in industrial engineering from the University of Zaragoza, Zaragoza, Spain, in 1989 and 1992, respectively. In 1994, he joined as an Associate Professor with the Departamento de Informatica e Ingenieria de Sistemas, University of Zaragoza, where he became a Full Professor in 2009 and also the Head Teacher. He was engaged in research on force and infrared sensors for robots. His current research interests include control systems and industry applications, computer vision, visual control, and multi-vehicle cooperative control.