# Dynamic max-consensus with local self-tuning

**D. Deplano** * **M. Franceschelli** * **A. Giua** *

* *Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Italy.*
*Emails:* {diego.deplano,mauro.franceschelli,giua}@unica.it

**Abstract:** This work describes a novel control protocol for multi-agent systems to solve the dynamic max-consensus problem. In this problem, each agent has access to an external time-varying signal and has the objective to estimate and track the maximum among all these signals by exploiting only local communications. The main strength of the proposed protocol is that it is able to self-tune its internal parameters in order to achieve an arbitrary small steady-state error without significantly affecting the convergence time. We also employ the proposed protocol in the context of distributed graph parameter estimations, such as size, diameter, and radius. We also provide simulations in the scenario of open multi-agent systems.

*Keywords:* Dynamic consensus, max-consensus, distributed estimation, open multi-agent systems.

## 1. INTRODUCTION

In the context of multi-agent systems, there has been a significant interest in the design of distributed algorithms to solve the so-called consensus or agreement problem.

**Problem of interest and motivation.** The consensus problem consists in the design of local interaction rule driving the agents to agree upon a common value of interest. Historically, such common value has been considered to be a function of the initial state of the agents. More more recently a variation of the problem that goes by the name of dynamic consensus problem has been considered, where the agents are required to converge to a state value which is a function of local time-varying reference signals fed to the agents. The current literature has paid major attention to the average value, for which Kia et al. (2019) have provided an insightful tutorial paper, but also other quantities have received some attention, such as the median value (Sanai Dashti et al., 2019; Vasiljevic et al., 2020), and the max/min value (Deplano et al., 2021b).

In a recent paper (Deplano et al., 2021b) we considered the previously unexplored dynamic max/min-consensus problem, which is also the focus of this paper. We propose an improved protocol for dynamic max/min-consensus that allows the designer to decouple convergence rate and steady-state error: this provides a way to jointly increase convergence rate and decrease steady-state error. Applications of max/min-consensus protocols are various and diversified, including monitoring and optimization (Iutzeler et al., 2012); distributed synchronization, such as time-synchronization (Dengchang et al., 2013) and target tracking (Petitti et al., 2011); network parameter estimation, such as cardinality (Lucchese et al., 2015), diameter and radius (Garin et al., 2012; Oliva et al., 2016; Deplano et al., 2021a), as well as highest/lowest node degree (Borsche and Attia, 2010).

**Related literature.** The standard max-consensus protocols can be traced back to the works of Tahbaz-Salehi and Jadbabaie (2006) and Cortés (2008), in continuous-time and discrete-time frameworks, respectively. Since then, different approaches and settings were considered, such as switching topologies (Nejad et al., 2010), second-order dynamics (Zhang and Li, 2018), asynchronous, delayed and noisy communications (Agrawal et al., 2019; Muniraju et al., 2019), gossip-based or randomized approaches (Iutzeler et al., 2012; Franceschelli and Gasparri, 2019), as well as open multi-agent systems (Abdelrahim et al., 2017), anonymous and resilient networks (Wang et al., 2018; Muniraju et al., 2019; Shang, 2020). The dynamic version of the max-consensus problem has recently been studied by us in (Deplano et al., 2021b), wherein two protocols are presented: one achieves bounded tracking and steady-state error, while the other one achieves zero steady-state error but requires a high memory burden.

**Main contributions.** In this paper, we provide a self-tuning version of the protocol we have previously proposed in (Deplano et al., 2021b) to solve the dynamic max-consensus problem. The main novel features are:

- Tracking and steady-state errors are decoupled by design;
- The protocol achieves bounded steady-state error that can be made arbitrarily small;
- The protocol achieves bounded tracking error that can be traded-off for improved convergence time;
- The memory burden does not increase with the dimension of the network.

⋆ This work was partially supported by the Fondazione Banco di Sardegna with the grant "Formal Methods and Technologies for the Future of Energy Systems", cup F72F20000350007.
D. Deplano is the corresponding author.

As a second contribution, we employ the proposed protocol in the context of open multi-agent systems in order to estimate and track some important time-varying parameters of the network, such as the number of agents, the radius, and the diameter. More precisely, we equip the algorithms proposed in (Deplano et al., 2021a) and (Deplano et al., 2021b) with the protocol proposed in this paper achieving the following improvements:

- The size-estimation technique employed in (Deplano et al., 2021b) is highly sensitive to the steady-state error of the dynamic max-consensus protocol, a problem that is completely solved by employing the protocol proposed in this paper.
- The radius/diameter estimation technique employed in (Deplano et al., 2021a) does make use of a static version of the max-consensus protocol, which requires a centralized step. The implementation of the novel protocol allows to avoid such a centralized step and enables the employment of the algorithm in the case of open multi-agent systems.

**Structure of the paper**. Section 2 presents the notation used throughout the paper along with some theoretical preliminaries. Section 3 presents our self-tuning dynamic max-consensus (STDMC) protocol. In Section 3.1 the dynamic max/min-consensus problem is formalized along with the main working assumptions. In Section 3.2 the proposed protocol is along with its theoretical characterization. In Section 4 we provide numerical simulations validating our results. Concluding remarks are given in Section 5.

## 2. NOTATION AND PRELIMINARIES

We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of real numbers and positive integer numbers, respectively. Maximum and minimum of a vector $v = [v_1, \ldots, v_m]^\intercal$, with $m \in \mathbb{N}$, are denoted by

$$\overline{v} = \max_{i=1,\ldots,m} v_i, \qquad \underline{v} = \min_{i=1,\ldots,m} v_i. \tag{1}$$

We consider networks modeled by undirected graphs $\mathcal{G} = (V, E)$, where $V = \{1, \ldots, n\}$ with $n \in \mathbb{N}$ is the set of *nodes*, and $E \subseteq V \times V$ is the set of *edges* connecting the nodes. The state and the input of the $i$-th agent are denoted by $x_i \in \mathbb{R}^m$ and $u_i \in \mathbb{R}$, respectively.

A path between two nodes $i, j \in V$ in a graph is a sequence of consecutive edges $\pi_{ij} = (i,p),(p,q),\ldots,(r,s),(s,j)$ where each successive edge shares a node with its predecessor. An undirected graph $\mathcal{G}$ is said to be *connected* if there exists a path $\pi_{ij}$ between any pair of nodes $i, j \in V$. The *diameter* $\delta_{\mathcal{G}}$ of a graph $\mathcal{G}$ is defined as the length (number of edges) of the longest shortest path between any pair of nodes in the graph.

Nodes $i$ and $j$ are *neighbors* if there exists an edge $(i,j) \in E$, which represents a point-to-point communication channel between nodes $i$ and $j$. The set of neighbors of the $i$-th node is denoted by $\mathcal{N}_i = \{j \in V : (i,j) \in E\}$. For sake of simplicity, we consider graphs without self-loops, i.e., $i \notin \mathcal{N}_i$, and define $\mathcal{N}_i^\circ = \mathcal{N}_i \cup \{i\}$.

## 3. SELF-TUNING DYNAMIC MAX-CONSENSUS (STDMC) PROTOCOL

### 3.1 *Problem statement and working assumptions*

Consider a network of $n$ agents modeled as discrete-time dynamical systems with scalar state $x_i \in \mathbb{R}$ for $i = 1, \ldots, n$. The $i$-th agent has access to a time-varying reference signal $u_i \in \mathbb{R}$ and interacts with other agents according to an undirected graph $\mathcal{G} = (V, E)$ and a local interaction rule

$$x_i(k) = f_i(u_i(k), x_j(k-1) : j \in \mathcal{N}_i^\circ). \tag{2}$$

The *dynamic max-consensus problem* consists in the design of proper local interaction rules $f_i(\cdot)$ for estimating and tracking the maximum $\overline{u}(k) \in \mathbb{R}$ among the time-varying reference signals. The performance can be expressed in terms of the convergence time and the tracking error

$$e(k) = \max_{i \in V} |x_i(k) - \overline{u}(k)|. \tag{3}$$

Our only assumption concerns the boundedness of the reference signals' variation, which is deemed a reasonable assumption in the dynamic consensus literature. We define the variation of the reference signals in one step as

$$\Delta u_i(k) = u_i(k) - u_i(k-1), \quad \forall i \in V. \tag{4}$$

and, in a similar way, we define the variation of the maximum among the reference signals as

$$\overline{\Delta u}(k) = \overline{u}(k) - \overline{u}(k-1). \tag{5}$$

*Assumption 1.* The maximum absolute variation of the reference signals in one step is bounded by a constant $\Pi \geq 0$, i.e.,

$$|\Delta u_i(k)| \leq \Pi, \quad \forall i \in V, \quad \forall k \geq 0. \tag{6}$$

### 3.2 *Proposed self-tuning protocol*

When the reference signals are assumed to be constant over time, i.e., $u(k) = u(0)$ for all $k \in \mathbb{N}$, the problem can be recast as a standard max-consensus problem by

$$x_i(k) = \max_{j \in \mathcal{N}_i^\circ} \{x_j(k-1)\}, \qquad x_i(0) = u_i(0), \tag{7}$$

which has been proved to converge in finite-time and with zero error (Tahbaz-Salehi and Jadbabaie, 2006). Instead, when the reference signals are assumed to be time-varying, the strategy we have previously proposed (Deplano et al., 2021b) modifies the above interaction rule into

$$x_i(k) = \max_{j \in \mathcal{N}_i^\circ} \{x_j(k-1) - \alpha, u_i(k)\}. \tag{8}$$

where $\alpha > 0$ is a design parameter. Such protocol has been proved to converge in finite-time and with bounded error, both depending on the parameter $\alpha$. More precisely, the parameter $\alpha$ trades-off tracking and steady-state errors for convergence time, with larger values of $\alpha$ leading to higher converges rate but also higher errors, and vice versa.

**STDMC Protocol :**
SELF-TUNING DYNAMIC MAX-CONSENSUS

**(Input):** Tuning parameters $\alpha^{\text{MAX}} \geq \alpha^{\text{MIN}} > 0$.
**(Initialization):** $x_i(0) \in \mathbb{R}$ for $i \in V$;
$\alpha_i(0) \in \{\alpha^{\text{MAX}}, \alpha^{\text{MIN}}\}$ for $i \in V$
**(Execution): for** $k = 1, 2, 3, \ldots$ **each node** $i$ **does**
Gather $x_j(k-1)$ and $\alpha_j(k-1)$ from each neighbor $j \in \mathcal{N}_i(k-1)$
Update the current state according to
$x_i(k) = \max_{j \in \mathcal{N}_i^\circ} \{x_j(k-1) - \alpha_j(k-1), u_i(k)\}$
Update the current parameter according to
$\alpha_i(k) = \begin{cases} \alpha^{\text{MAX}} & \text{if } x_i(k) < x_i(k-1) \\ \alpha^{\text{MIN}} & \text{otherwise} \end{cases}$

The strategy proposed in this paper is that of equipping the update rule in eq. (8) with local and self-tuning parameters $\alpha_i(k)$ for $i \in V$, as follows

$$x_i(k) = \max_{j \in \mathcal{N}_i^\circ} \left\{x_j(k-1) - \alpha_j(k-1), u_i(k)\right\}, \quad (9)$$

Intuitively, one can expect that the parameters $\alpha_i$ should take large values when the maximum input varies and the tracking task becomes the priority, while they should take arbitrarily small values when the maximum input remains constant and the estimation accuracy becomes more important. This is correct, and we further clarify this qualitative reasoning by discussing the following two cases.:

- The maximum input $\overline{u}$ is higher than all states $x_i$. In this case, the agent $i$ with the maximum input $u_i = \overline{u}$ updates its state to $\overline{u}$, regardless of the value of $\alpha_i$. Thus, the parameter $\alpha_i$ can be an arbitrarily small value. Note that in this case, the state of the agent $i$ increases (or remains the same) after the update.
- The maximum input $\overline{u}$ is lower than all states $x_i$. In this case, each agent $i$ update its state to the maximum among $x_j - \alpha_j$ with $j \in \mathcal{N}_i$, which depends on the parameters $\alpha_j$. Thus, the parameters $\alpha_j$ must be sufficiently large in order to guarantee the tracking of the maximum input. Note that in this case, the state of the agents eventually decreases if the maximum input continues decreasing. This process stops when the input remains constant (or increases) and the first case holds instead.

As a consequence of the above discussion, we propose to make the parameters $\alpha_i$ switch between two design parameters $\alpha^{\text{MAX}} \geq \alpha^{\text{MIN}} > 0$ according to

$$\alpha_i(k) = \begin{cases} \alpha^{\text{MAX}} & \text{if } x_i(k) < x_i(k-1) \\ \alpha^{\text{MIN}} & \text{otherwise} \end{cases}. \quad (10)$$

Due to the above update rule, if the state of agent $i$ is decreasing, then its local parameter $\alpha_i$ is set to a high value $\alpha^{\text{MAX}}$ in order to speed up the convergence toward the maximum input that has a smaller value. Contrarily, if the state of agent $i$ is increasing or constant, then its local parameter $\alpha_i$ is set to a small value $\alpha^{\text{MIN}}$ in order to improve the estimation accuracy.

We describe the steps required to implement the interaction rule (9) with the local self-tuning (10) in the STDMC Protocol, shown in the column on the left, while the next theorem characterizes its convergence time and tracking error.

*Theorem 1.* Consider a multi-agent system executing the STDMC Protocol under Assumption 1 and let $k = 0$ be the initial time. If graph $\mathcal{G}$ is connected and if

$$\alpha^{\text{MAX}} > \Pi, \quad (11)$$

then there exists a convergence time $T_c \geq 0$ such that the tracking error is bounded for $k \geq T_c$ by

$$e(k) \leq (\alpha^{\text{MAX}} + \Pi)\delta_{\mathcal{G}}, \quad (12)$$

where $\delta_{\mathcal{G}}$ is the diameter of graph $\mathcal{G}$, and it holds

$$T_c \leq \delta_{\mathcal{G}} + \max\left\{\frac{\overline{x}(\delta_{\mathcal{G}}) - \overline{u}(\delta_{\mathcal{G}})}{\alpha^{\text{MAX}} - \Pi}, 0\right\}. \quad (13)$$

**Proof sketch of Theorem 1.** It can be shown[1] that there exists a convergence time $T_c \geq 0$ such that the maximum and minimum among the agents' state for $k \geq T_c$ are bounded by the following

$$\overline{x}(k) \leq \overline{u}(k) + (\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}} \quad (14)$$
$$\underline{x}(k) \geq \overline{u}(k) - (\Pi + \alpha^{\text{MAX}})\delta_{\mathcal{G}}. \quad (15)$$

The bound (14) comes from the consideration that if $\overline{u}(k) \geq \overline{x}(k)$, then due to the update rule in eq. (9) it must hold $\overline{x}(k+1) = \overline{u}(k+1)$. otherwise, if the $\overline{u}(k) < \overline{x}(k)$ then their distance at subsequent times depends on their maximum variations. In the worst case, the maximum state decreases by $\alpha^{\text{MIN}}$ for $\delta_{\mathcal{G}}$ steps and the maximum input decreases by $\Pi$, leading to the bound in eq. (14). Indeed, after $\delta_{\mathcal{G}}$ steps, all the parameters $\alpha_i$ are update to $\alpha^{\text{MAX}}$ according to the self-tuning in eq. (10), and thus the distance becomes smaller after $\delta_{\mathcal{G}}$ steps. The bound (15) comes from the consideration that the maximum distance between the minimum state $\underline{x}$ and the maximum input $\overline{u}$ occurs when during the tracking all $\alpha_i$ are set to $\alpha^{\text{MAX}}$ and the input suddenly reverses behavior and start increasing with maximum variation $\Pi$. Thus, their maximum distance is proportional to $\Pi + \alpha^{\text{MAX}}$, but after at most $\delta_{\mathcal{G}}$ steps the $\alpha_i$ are set to $\alpha^{\text{MIN}}$ according to the self-tuning in eq. (10), and thus the distance becomes smaller after $\delta_{\mathcal{G}}$ steps.

Note that the convergence time $T_c$ is upper bounded by the diameter of the network, which is the largest time needed by the network to complete the cascade update. Indeed, $\delta_{\mathcal{G}}$ is the length of the longest shortest path between any pair of agents. Now, at time $k = \delta_{\mathcal{G}}$ two cases may occur:

- $\overline{x}(\delta_{\mathcal{G}}) \leq \overline{u}(\delta_{\mathcal{G}})$. In this case, both bounds in eqs. (14)-(15) hold, and thus $T_c \leq \delta_{\mathcal{G}}$.
- $\overline{x}(\delta_{\mathcal{G}}) > \overline{u}(\delta_{\mathcal{G}})$. In this case, the maximum state $\overline{x}$ decreases with rate $\alpha^{\text{MAX}}$ and, in the worst case, the maximum input decreases with rate $\Pi$. Due to eq. (11), then there exists a time $T'$ such that $\overline{x}(\delta_{\mathcal{G}} + T') \leq \overline{u}(\delta_{\mathcal{G}} + T')$ and $T_c \leq \delta_{\mathcal{G}} + T'$, given by

$$T' = \max\left\{\frac{\overline{x}(\delta_{\mathcal{G}}) - \overline{u}(\delta_{\mathcal{G}})}{\alpha^{\text{MAX}} - \Pi}, 0\right\}. \quad (16)$$

[1] For the convenience of the reviewers, formal proofs of the bounds in eqs. (14)-(15)-(16) are given in Appendix A, which is available at (Deplano et al., 2022).

From the above discussion one can verify the upper bound to the convergence time given in eq. (13). Now, exploiting the upper bound in eq. (14) and the lower bound in eq. (15), the bound on the tracking error can be derived for $k \geq \delta_{\mathcal{G}} + T'$ as follows

$$\begin{aligned} e(k) &= \max_{i \in V} |x_i(k) - \overline{u}(k)| \\ &= \max\{|\overline{x}(k) - \overline{u}(k)|, |\underline{x}(k) - \overline{u}(k)|\} \\ &\leq \max\{(\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}}, (\Pi + \alpha^{\text{MAX}})\delta_{\mathcal{G}}\} \\ &\leq (\Pi + \alpha^{\text{MAX}})\delta_{\mathcal{G}}, \end{aligned}$$

completing the proof. □

The characterization provided in Theorem 1 reveals that the convergence time is inversely proportional to $\alpha^{\text{MAX}}$, while the tracking error is directly proportional to $\alpha^{\text{MAX}}$. Therefore, $\alpha^{\text{MAX}}$ trades off convergence time for tracking error. On the other hand, the parameter $\alpha^{\text{MIN}}$ does not affect either, but it does affect the steady-state error in the case of constant inputs, as formalized next.

*Theorem 2.* Consider a generic time $k_0 \geq T_c$. The estimation error for Theorem 1, in the case all reference signals remain constant for $k \geq k_0$, satisfies the next condition

$$e(k) \leq \alpha^{\text{MIN}}\delta_{\mathcal{G}}, \qquad k \geq k_0 + 2\delta_{\mathcal{G}}. \tag{17}$$

**Proof.** By Theorem 1, it holds

$$\overline{u}(k) \leq \overline{x}(k) \leq \overline{u}(k) + (\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}}, \qquad k \geq T_c.$$

From the proof of Theorem 1, if the above upper bound holds strictly at $k$, i.e., $\overline{x}(k) = \overline{u}(k) + (\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}}$, then:

- All $\alpha_i(k)$ are exactly equal to $\alpha^{\text{MAX}}$;
- $\overline{u}(k)$ is decreasing, i.e., $\overline{u}(k) < \overline{u}(k - \delta_{\mathcal{G}})$.

In the worst-case scenario, this is the case at $k = k_0$, since then the inputs remain constant and equal to $u(k_0)$. Therefore, the maximum input starts decreasing at each time step by a factor $\alpha^{\text{MAX}}$, until the maximum input $\overline{u}(k_0)$ is reached. This process takes exactly $\delta_{\mathcal{G}}$ steps, as it is shown next,

$$\begin{aligned} \overline{x}(k_0 + \ell) = \ & \overline{x}(k_0) - \ell\alpha^{\text{MAX}} && \leq \overline{u}(k_0) \\ \overline{x}(k_0 + \ell) = \ & \overline{u}(k_0) + (\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}} - \ell\alpha^{\text{MAX}} && \leq \overline{u}(k_0) \\ & (\Pi - \alpha^{\text{MIN}})\delta_{\mathcal{G}} && \leq \ell\alpha^{\text{MAX}} \\ & \frac{\Pi - \alpha^{\text{MIN}}}{\alpha^{\text{MAX}}}\delta_{\mathcal{G}} && \leq \ell \end{aligned}$$

and since $(\Pi - \alpha^{\text{MIN}})/\alpha^{\text{MAX}} \in (0, 1)$, it follows that the smallest value satisfying the above inequality is $\ell = \delta_{\mathcal{G}}$.

Now, once $\delta_{\mathcal{G}}$ steps have elapsed, the parameter of the agent with the maximum input is updated to $\alpha^{\text{MIN}}$ due to the self-tuning rule in eq. (10) for $k \geq k_0 + \delta_{\mathcal{G}}$. Due to the local interaction rule in eq. (9), its 1-hop neighbors update their state to either $\overline{x}(k) - \alpha^*$ with $\alpha^* \in \{\alpha^{\text{MIN}}, \alpha^{\text{MAX}}\}$ or to their own input $u_i$. In both cases, according to the update rule in eq. (10), at subsequent instant of times their local parameter is updated to the value of $\alpha^{\text{MIN}}$. The cascade effect updates all parameters $\alpha_i$ with $i \in V$ to $\alpha^{\text{MIN}}$ in at most $\delta_{\mathcal{G}}$ steps. We conclude that

$$\overline{u}(k_0) \geq \overline{x}(k) > \underline{x}(k) = \overline{u}(k) - \alpha^{\text{MIN}}\delta_{\mathcal{G}}, \quad k \geq k_0 + 2\delta_{\mathcal{G}},$$

from which the statement of the theorem follows. □

## 4. NUMERICAL SIMULATIONS

We test the results for the STDMC Protocol by considering the worst-case scenario of a network with line topology, which maximizes the number of steps needed to make the information flow through the network, that is $\delta_{\mathcal{G}} = n - 1$ steps, thus maximizing the bound on both the tracking and the steady-state errors.

### *4.1 Example 1: convergence time and steady-state error*

We show in Fig. 1 the time-evolution of a network with $n = 10$ agents in a line configuration, with diameter $\delta_{\mathcal{G}} = 9$, executing the STDMC Protocol. The agents are uniformly initialized within $[0, 1]$, the reference signals remain constant and equal to 0, except for the 6-th input that is initialized at $u_6(0) = 0.5$ and varies according to

$$u_6(k+1) = \begin{cases} u_6(k) - \Pi & \text{if } k \in [100, 110) \\ u_6(k) & \text{otherwise} \end{cases}.$$

where $\Pi = 0.02$ is the maximum absolute variation as in Assumption 1. The parameters of the STDMC Protocol are chosen accordingly to Theorem 1 and are given by

$$\alpha^{\text{MAX}} = 0.022 > \Pi, \quad \alpha^{\text{MIN}} = 0.005.$$

This example allows to verify the validity of the bound $\varepsilon_{ss}$ on the steady-state error given Theorem 2. Indeed, all inputs remain constant for $k \geq 110$ and the bound $\varepsilon_{ss} = \alpha^{\text{MIN}}\delta_{\mathcal{G}} = 0.045$ is satisfied for $k \geq k_0 + 2\delta_{\mathcal{G}} = 128$.

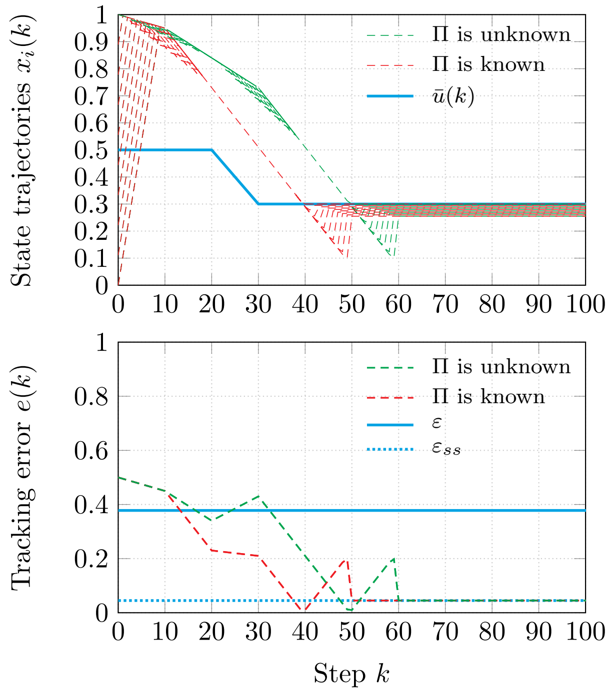


Fig. 1. *Example 1*: Evolution of a network with line topology running the STDMC Protocol. $\varepsilon$ and $\varepsilon_{ss}$ are the bound on the tracking and steady-state error.

### 4.2 *Example 2: tracking error and comparison with non self-tuning dynamic max-consensus*

We compare in Figs. 2-3 the time-evolution of a network with $n = 10$ agents in a line configuration executing the STDMC Protocol with self-tuning and without self-tuning. The agents are uniformly initialized in the interval $[0, 1.5]$ and all reference signals remain constant and equal to $-1$, except for the 6-th agent. The input of the agent 6 is initialized at $u_6(0) = 0.5$ and varies according to

$$u_6(k+1) = \begin{cases} u_6(k) - \Pi & \text{if } k \in [100, 150) \\ u_6(k) + \Pi & \text{if } k \in [200, 250) \\ u_6(k) & \text{otherwise} \end{cases} .$$

where $\Pi = 0.02$ is the maximum absolute variation as in Assumption 1. In Fig. 2, we show the estimation result provided by the STDMC Protocol when the self-tuning is designed, accordingly to Theorem 1, as shown next

$$\alpha^{\text{MAX}} = 0.022 > \Pi, \quad \alpha^{\text{MIN}} = 10^{-12}.$$

Instead, in Fig. 2, we show the estimation result provided by the STDMC Protocol when the self-tuning is disabled, which amounts to the DMC Protocol previously proposed by us in (Deplano et al., 2021b) with constant parameter $\alpha = 0.022$. It is straightforward to notice how the self-tuning logic improves the accuracy of the estimation, without affecting the convergence time. By looking at Fig. 2, one can also validate the characterization of the STDMC Protocol provided in Theorems 1-2. In particular, it can be verified that the tracking error is always bounded by $\varepsilon = (\Pi + \alpha^{\text{MAX}})\delta_{\mathcal{G}} = 0.378$, and, more importantly, that the steady state error is almost nullified by the choice $\alpha^{\text{MIN}} = 10^{-12}$ at $k \in [66, 100] \cup [209, 300]$.

### 4.3 *Example 3: graph parameters estimation in open multi-agent systems*

In the third simulation shown in Fig. 4, we employ the proposed protocol as a subroutine of the algorithms presented in our previous works (Deplano et al., 2021a,b) for the estimation and track of some important graph parameters in open multi-agent systems, wherein the agents may join and leave as time goes by. For the convenience of the reviewers, we detail both these algorithms in the unified SDR Protocol presented in Appendix B, which is available at (Deplano et al., 2022). In particular, the algorithms enable to track the following time-varying parameters:

- Size $s(k)$: the number of agents within the network;
- Radius $d(k)$: The length of the minimum distance between any pair of agents in the network;
- Diameter $d(k)$: The length of the maximum distance between any pair of agents in the network.

We consider a network of $n \leq 100$ agents that are allowed to join and leave the network every $\Upsilon = 200$ time steps, as well as establishing or closing communications with other agents. This leads to a network with a time-varying number of nodes, as well as time-varying diameter and radius. We denote by $\hat{s}(k)$, $\hat{d}(k)$, $\hat{r}(k)$ their estimations. The parameters are set as shown next

$$\alpha^{\text{MAX}} = 10^{-1}, \quad \alpha^{\text{MIN}} = 10^{-12} \approx 0.$$

Fig. 4 (top) shows the estimation $\hat{s}(k)$ (red dashed curve) of the network's size $s(k)$ (blue solid curve), showing how the algorithm is capable of tracking changes of the time-varying size, with fast convergence rate ruled by parameter $\alpha^{\text{MAX}}$ and high accuracy ruled by $\alpha^{\text{MIN}}$.

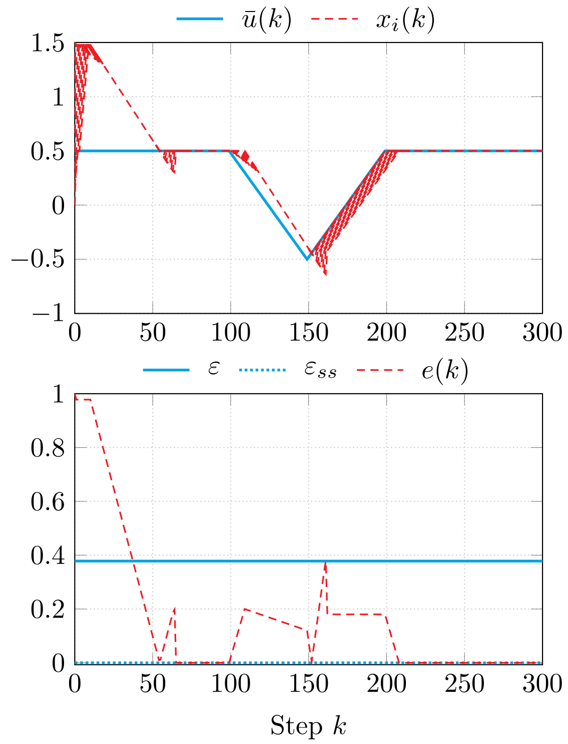


Fig. 2. *Example 2*: Evolution of the network in Example 2 running the STDMC Protocol.

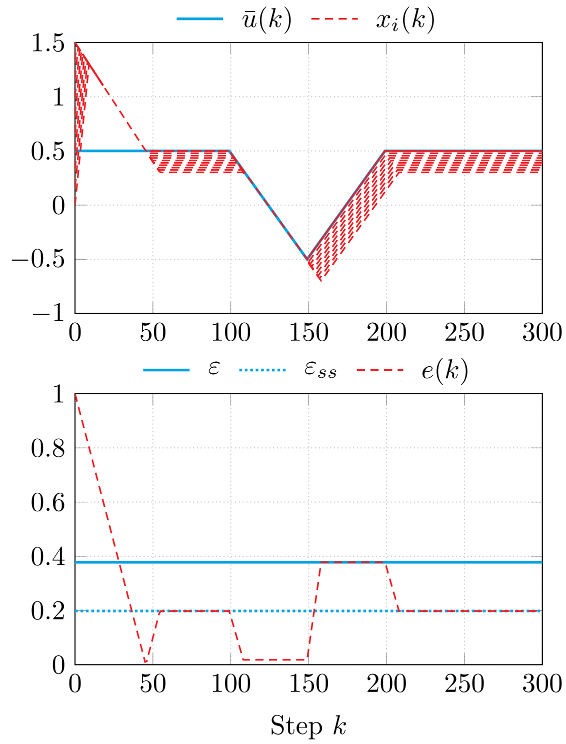


Fig. 3. *Example 2*: Evolution of the network in Example 2 running the STDMC Protocol without self-tuning.

We also show the estimation when the non self-tuning dynamic max-consensus (DMC) Protocol proposed in (Deplano et al., 2021b) is employed (green dashed curve), which amounts to select $\alpha^{\text{MAX}} = \alpha^{\text{MIN}} = \alpha^*$. In order to provide a fair comparison, we select the good trade-off $\alpha^* = 10^{-4}$. The comparison reveals that the employment of the STDMC Protocol allows achieving both higher convergence rate and accuracy. Fig. 4 (bottom) shows the estimations $\hat{d}(k), \hat{e}(k)$ (red dashed curve) of the network's diameter and radius $d(k), r(k)$ (blue solid curve), respectively. Since the diameter is always greater or equal to the radius, the curves on top refer to the diameter estimation while the curves below refer to the radius estimation. Note that we cannot provide a comparison with the original algorithm proposed in (Deplano et al., 2021a), since it cannot be employed in an open network setting without the self-tuning logic proposed in this paper.

## 5. CONCLUSIONS

We have proposed the self-tuning dynamic max-consensus (STDMC) Protocol, which enables the agents to track the time-varying maximum value of a set of reference signals given as inputs to the agents. As the name suggests, it is capable of self-tune some internal parameters in order to minimize both tracking and steady-state errors, which are decoupled by design. We further provided simulations when the STDMC Protocol is employed as a subroutine of two state-of-art algorithms to track size, the diameter, and the radius of the network in open and time-varying multi-agent systems.

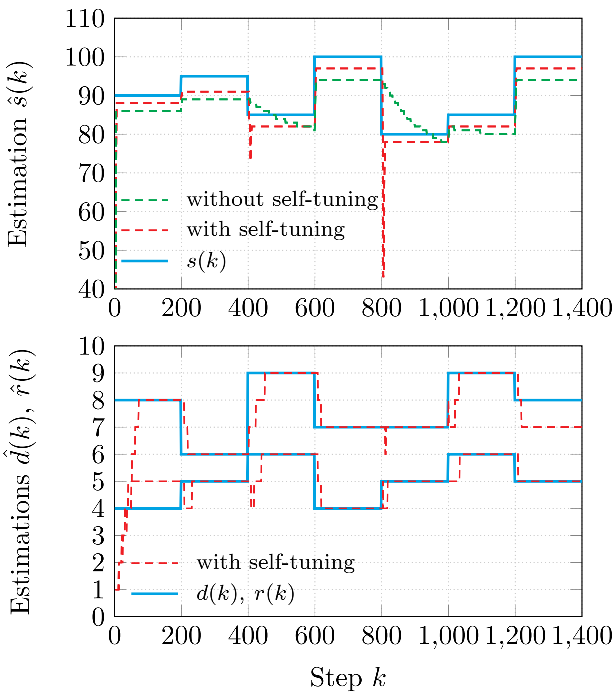


Fig. 4. *Example 3*: Estimation and tracking of the size (top) and radius/diameter (bottom) in an open multi-agent system running the protocols proposed in (Deplano et al., 2021a,b) implementing the STDMC Protocol as a subroutine. NB: the green curve shows the estimation without the self-tuning logic.

## REFERENCES


Abdelrahim, M., Hendrickx, J., and Heemels, W. (2017). Max-consensus in open multi-agent systems with gossip interactions. In *56th IEEE Conference on Decision and Control (CDC)*, 4753–4758.

Agrawal, N., Frey, M., and Stańczak, S. (2019). A scalable max-consensus protocol for noisy ultra-dense networks. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 1–5.

Borsche, T. and Attia, S.A. (2010). On leader election in multi-agent control systems. In *22th Chinese Control and Decision Conference (CCDC)*, 102–107.

Cortés, J. (2008). Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3), 726–737.

Dengchang, Z., Zhulin, A., and Yongjun, X. (2013). Time synchronization in wireless sensor networks using max and average consensus protocol. *International Journal of Distributed Sensor Networks*, 9(3), 192128–1/10.

Deplano, D., Franceschelli, M., and Giua, A. (2021a). Distributed tracking of graph parameters in time-varying anonymous networks. In *60th IEEE Conference on Decision and Control (CDC)*, 6258–6263.

Deplano, D., Franceschelli, M., and Giua, A. (2021b). Dynamic min and max consensus and size estimation of anonymous multi-agent networks. *IEEE Transactions on Automatic Control.*

Deplano, D., Franceschelli, M., and Giua, A. (2022). Dynamic max-consensus with local self-tuning. URL `https://unica.it/unica/protected/361711/0/def/ref/DAT258110/`.

Franceschelli, M. and Gasparri, A. (2019). Multi-stage discrete time and randomized dynamic average consensus. *Automatica*, 99, 69–81.

Garin, F., Varagnolo, D., and Johansson, K.H. (2012). Distributed estimation of diameter, radius and eccentricities in anonymous networks. In *IFAC Proceedings Volumes*, volume 45, 13–18. Elsevier.

Iutzeler, F., Ciblat, P., and Jakubowicz, J. (2012). Analysis of max-consensus algorithms in wireless channels. *IEEE Transactions on Signal Processing*, 60(11), 6103–6107.

Kia, S., Van Scoy, B., Cortes, J., Freeman, R., Lynch, K., and Martinez, S. (2019). Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems*, 39(3), 40–72.

Lucchese, R., Varagnolo, D., Delvenne, J., and Hendrickx, J. (2015). Network cardinality estimation using max consensus: The case of bernoulli trials. In *54th IEEE Conference on Decision and Control*, 895–901.

Muniraju, G., Tepedelenlioglu, C., and Spanias, A. (2019). Analysis and design of robust max consensus for wireless sensor networks. *IEEE Transactions on Signal and Information Processing over Networks*, 5(4), 779–791.

Nejad, B.M., Attia, S.A., and Raisch, J. (2010). Max-consensus in a max-plus algebraic setting: The case of switching communication topologies. In *IFAC Proceedings Volumes*, volume 43, 173–180.

Oliva, G., Setola, R., and Hadjicostis, C.N. (2016). Distributed finite-time calculation of node eccentricities, graph radius and graph diameter. *Systems & Control Letters*, 92, 20–27.

Petitti, A., Di Paola, D., Rizzo, A., and Cicirelli, G. (2011). Consensus-based distributed estimation for target tracking in heterogeneous sensor networks. In *50th IEEE Conference on Decision and Control (CDC) and European Control Conference (ECC)*, 6648–6653.

Sanai Dashti, Z.A., Seatzu, C., and Franceschelli, M. (2019). Dynamic consensus on the median value in open multi-agent systems. In *58th IEEE Conference on Decision and Control (CDC)*, 3691–3697.

Shang, Y. (2020). Resilient consensus in multi-agent systems with state constraints. *Automatica*, 122, 109288.

Tahbaz-Salehi, A. and Jadbabaie, A. (2006). A one-parameter family of distributed consensus algorithms with boundary: From shortest paths to mean hitting times. In *45th IEEE Conference on Decision and Control (CDC)*, 4664–4669.

Vasiljevic, G., Petrovic, T., Arbanas, B., and Bogdan, S. (2020). Dynamic median consensus for marine multi-robot systems using acoustic communication. *IEEE Robotics and Automation Letters*, 5(4), 5299–5306.

Wang, A., Mu, N., and Liao, X. (2018). Min-max consensus algorithm for multi-agent systems subject to privacy-preserving problem. In *International Conference on Neural Information Processing*, 132–142.

Zhang, Y. and Li, S. (2018). Second-order min-consensus on switching topology. *Automatica*, 96, 293–297.