Dynamic Max-Consensus in Multi-Agent Systems and Graph Parameter Estimation in Open Networks

Diego Deplano*, **Mauro Franceschelli***, **Alessandro Giua***

*Department of Electrical and Electronic Engineering, University of Cagliari, Italy

1^{TH} *Seminar – 8 March 2023*

Outline

Outline

1. Multi-Agent Systems and the Consensus Problem
2. Dynamic Max-Consensus Protocols
3. Graph parameter estimation in open networks
4. Ongoing work: consensus via distributed optimization
5. Future perspectives

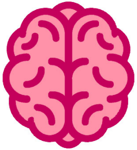
What is an agent?

"Agents are **computational** *systems that inhabit some complex dynamic environment, sense and act* **autonomously** *in this environment, and by doing so realize a set of* **goals** *or tasks for which they are designed."*

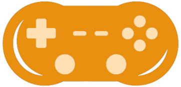
P. Maess, "*Artificial life meets entertainment: Life like autonomous agents*", Communications of the ACM, 1995.

What is an agent? Some examples

Self-driving car



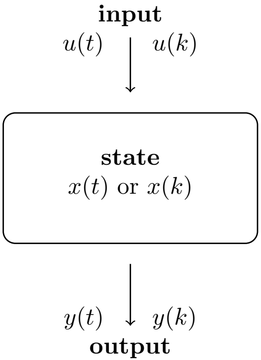
Human's opinion



Player

Computational unit

Model of the agent: dynamical system



An agent is modeled as a dynamical system with a state-space representation:

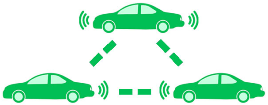
- Continuous time (CT): $\dot{x}(t) = f(y(t), u(t))$ and $y(t) = h(x(t))$ with $t \in \mathbb{R}$;
- Discrete time (DT): $x(k+1) = f(y(k), u(k))$ and $y(k) = h(x(k))$ with $k \in \mathbb{N}$.

What is a multi-agent system?

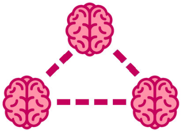
"A multi-agent system is a **coupled** *network of agents that* **work together** *to find answers to problems that are* **beyond** *the individual capabilities or knowledge of each agent."*

P. Stone, M. Veloso, "Multiagent systems: A survey from a machine learning perspective", Autonomous Robots, 2000.

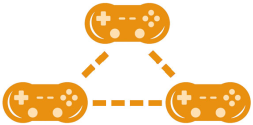
What is a multi-agent system? Some examples



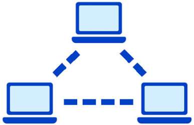
Multi-Robot control



Social opinion dynamics

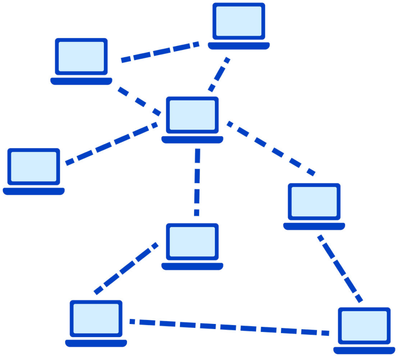


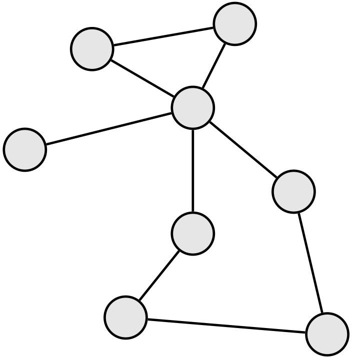
Game theory



Optimization theory

Model of the network: graph





A network of agents is modeled with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where:

- $\mathcal{V} \subset \mathbb{N}$ is the set of nodes (grey circles) modeling the agents;
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (black lines) modeling the flow information among agents.

Notation and working assumptions

We consider Multi-Agents Systems (MASs) under the following working assumptions:

- Discrete time framework: $k \in \mathbb{N}$;
- A number of agents equal to $n \in \mathbb{N}$;
- Scalar agents: $x_i(k) \in \mathbb{R}$;
- Time-varying scalar inputs: $u_i(k) \in \mathbb{R}$ for all $k \in \mathbb{N}$;
- Identity output map: $y_i(k) = x_i(k)$ for all $k \in \mathbb{N}$;
- Undirected interaction graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$;
- Set of neighbors $\mathcal{N}_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ for agents $i \in \mathcal{V}$;
- Local interaction protocols $f_i : \mathbb{R}^n \to \mathbb{R}$ for agents $i \in \mathcal{V}$;

Thus, the dynamics of each agent is

$$x_i(k+1) = f_i(u_i(k), x_i(k), x_j(k) : j \in \mathcal{N}_i), \quad \forall i \in \mathcal{V}, k \in \mathbb{N}, \tag{1}$$

and $x(k) = [x_1(k), \ldots, x_n(k)]^\top \in \mathbb{R}^n$ denotes the state of the network.

The finite-time consensus problem

Definition: Static consensus problem

The static consensus problem consists in the design of a set of local interaction protocols f_i such that each agent converges to a function $g : \mathbb{R}^n \to \mathbb{R}$ of the initial agents' states $x(0) = [x_1(0), \ldots, x_n(0)]^\top \in \mathbb{R}^n$, i.e., there are $k^*, \varepsilon \geq 0$ such that

$$|x_i(k) - g(x(0))| < \varepsilon, \qquad k \geq k^*, \quad \forall i \in \mathcal{V}.$$

Definition: Dynamic consensus problem

The dynamic consensus problem consists in the design of a set of local interaction protocols f_i such that each agent converges to a function $g : \mathbb{R}^n \to \mathbb{R}$ of the time-varying inputs $u(k) = [u_1(k), \ldots, u_n(k)]^\top \in \mathbb{R}^n$, i.e., there are $k^*, \varepsilon \geq 0$ such that

$$|x_i(k) - g(u(k))| < \varepsilon, \qquad k \geq k^*, \quad \forall i \in \mathcal{V}.$$

Literature: dynamic consensus on the average function

The **average** is the *sum of values of a data set divided by the number of values*, which is denoted by:

$$\operatorname{avg}(u(k)) = \frac{1}{n}\sum_{i\in\mathcal{V}} u_i(k).$$

- Spanos, Olfati-Saber, and Murray, "Dynamic consensus on mobile networks", in *IFAC World Congr.* (2005)
- Freeman, Yang, and Lynch, "Stability and convergence properties of dynamic average consensus estimators", in *IEEE 45th Conf. on Dec. and Control (2006)*
- Zhu and Martinez, "Discrete-time dynamic average consensus", in *Automatica* (2010)
- Chen, Cao and Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives", in *IEEE Trans. Autom. Control* (2012).
- Kia, Cortés, and Martinez "Dynamic average consensus under limited control authority and privacy requirements", in *Int. Journal of Robust and Nonlin. Control* (2015)
- Scoy, Freeman, and Lynch, "A fast robust nonlinear dynamic average consensus estimator in discrete time", in *5th IFAC NecSys* (2015)
- Franceschelli, and Gasparri, "Multi-stage discrete time and randomized dynamic average consensus", in *Automatica* (2019)
- George and Freeman, "Robust dynamic average consensus algorithms", in *IEEE Trans. Autom. Control* (2019)
- Montijano E. and J.I., Sagues, and Martinez, "Robust discrete time dynamic average consensus", in *IEEE Trans. Autom. Control* (2019)
- Kia, Scoy, Cortés, Freeman, Lynch and Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms", in *IEEE Control Systems Magazine* (2019).

Literature: dynamic consensus on the median function

The **median** is the *middle value separating the greater and lesser halves of a data set*, which, letting $u_1(k), \ldots, u_n(k)$ be sorted in ascending order, is denoted by:

$$\mathrm{med}(u(k)) = \begin{cases} u_{\frac{n+1}{2}}(k) & \text{if } n \text{ is odd} \\ \frac{1}{2}\left(u_{\frac{n}{2}}(k) + u_{\frac{n}{2}+1}(k)\right) & \text{if } n \text{ is even} \end{cases}.$$

- Sanai Dashti, Seatzu, and Franceschelli, "Dynamic consensus on the median value in open multi-agent systems", in *IEEE 58th Conf. on Dec. and Control* (2019).
- Vasiljevic, Petrovic, Arbanas, and Bogdan, "Dynamic median consensus for marine multi-robot systems using acoustic communication", in *IEEE Robot. and Autom. Lett.* (2020).
- Yu, Chen and Kar, "Dynamic median consensus over random networks", in *IEEE 60th Conf. on Dec. and Control* (2021).

Literature: dynamic consensus on the max function

The **maximum** is the *highest value of a data set*, which is denoted by:

$$\max(u(k)) = \max_{i \in \mathcal{V}} u_i(k).$$

- Deplano, Franceschelli, and Giua, "Dynamic min and max consensus and size estimation of anonymous multi-agent networks", in *IEEE Trans. Autom. Control* (2023).
- Deplano, Franceschelli, and Giua, "Dynamic max–consensus with local self–tuning", in *IFAC–PapersOnLine* (2022).
- Deplano, Franceschelli, and Giua, "Distributed tracking of graph parameters in anonymous networks with time–varying topology", in *IEEE Conference on Decision and Control*, (2021).
- Deplano, Franceschelli, Giua, "Discrete-time Dynamic consensus on the max value", in *15th European Workshop on Advanced Control and Diagnosis*, Springer (2021)
- Lippi, Furchi, Marino, and Gasparri, "An adaptive distributed protocol for finite–time infimum or supremum dynamic consensus", in *IEEE Control Systems Letters* (2023).
- Sen, Sahoo, and Slingh, "Global max-tracking of multiple time-varying signals using a distributed protocol", in *IEEE Control and Sys. Lett.* (2022)

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)

- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)

- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)

- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)

- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)

- and many others...

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)

- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)

- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)

- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)

- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)

- and many others...

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)

- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)

- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)

- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)

- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)

- and many others...

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)
- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)
- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)
- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)
- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)
- and many others...

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)

- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)

- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)

- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)

- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)

- and many others...

Applications

Possible applications range across different fields:

- Real-time monitoring in decentralized systems;

  Simpson-Porco and Bullo, "Distributed monitoring of voltage collapse sensitivity indices", in *IEEE Trans. Smart Grid* (2016)

- Network's parameter estimation in anonymous networks;

  Garin, Varagnolo, and Johansson, "Distributed estimation of diameter, radius and eccentricities in anonymous networks", in *3rd IFAC NecSys* (2012)

  Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2013)

- Online optimization in distributed systems;

  Jiang and Charalambous, "Distributed ADMM using finite-time exact ratio consensus in digraphs", in *European Control Conf.* (2021)

  Bastianello and Carli, "ADMM for Dynamic Average Consensus Over Imperfect Networks", in *9th IFAC Necsys* (2022)

- Distributed synchronization in wireless sensor networks;

  Z. Dengchang *et al.*, "Time synchronization in wireless sensor networks using max and average consensus protocol" (2013)

- Leader election in multi-agent systems;

  T. Borsche and S. A. Attia, "On leader election in multi-agent control systems" (2010)

- and many others...

Outline

Static max-consensus protocol

Problem: The agents want to estimate the maximum among their initial values, i.e.,

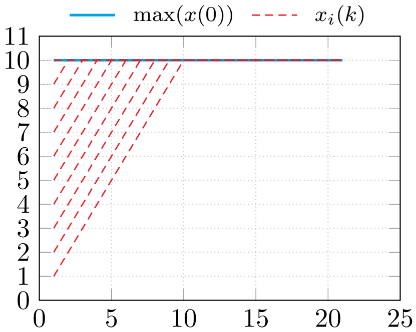
$$\max(x(0)) = \max_{i \in \mathcal{V}} x_i(0).$$

The most popular max-consensus protocol is the following

$$x_i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{x_j(k-1)\}, \qquad i \in \mathcal{V}.$$

Example: network with $n = 10$ agents, with initial conditions:

$$x_1(0) = 1$$
$$x_2(0) = 2$$
$$\vdots$$
$$x_{10}(0) = 10$$



A naive generalization to the dynamic problem

Problem: The agents want to track the maximum among time-varying inputs, i.e.,

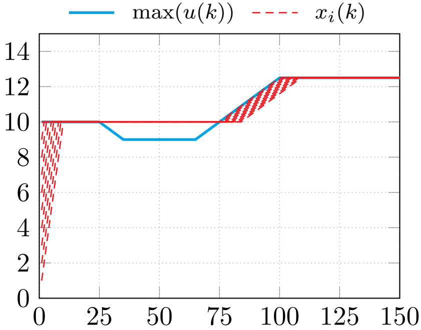
$$\max(u(k)) = \max_{i\in\mathcal{V}} u_i(k).$$

A naive generalization is the following

$$x_i(k) = \max_{j\in\mathcal{N}_i\cup\{i\}} \{x_j(k-1), u_i(k)\}, \qquad i \in \mathcal{V}.$$

Example: network with $n = 10$ agents, with initial conditions:

$$x_1(0) = u_1(0) = 1$$
$$x_2(0) = u_2(0) = 2$$
$$\vdots$$
$$x_{10}(0) = u_{10}(0) = 10$$



Our proposed protocols

- The proportional dynamic consensus protocol (PDMC) is

$$x_i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{\alpha \cdot x_j(k-1), u_i(k)\}, \qquad \alpha \in (0,1),\ i \in \mathcal{V},$$

Deplano, Franceschelli, Giua, "Discrete-time Dynamic consensus on the max value", in *15th European Workshop on Advanced Control and Diagnosis*, Springer (2021)

- **The dynamic consensus protocol (DMC) is**

$$x_i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{x_j(k-1) - \alpha, u_i(k)\}, \qquad \alpha \geq 0,\ i \in \mathcal{V}.$$

Deplano, Franceschelli, and Giua, "Dynamic min and max consensus and size estimation of anonymous multi-agent networks", in *IEEE Trans. Autom. Control* (20223).

- The exact dynamic consensus protocol (EDMC) is

$$\begin{aligned} x_i^0(k) &= u_i(k) \\ x_i^\ell(k) &= \max_{j \in \mathcal{N}_i \cup \{i\}} \{x_j(k-1)\}, \quad \ell = 1, \dots, m' \end{aligned} \qquad m \geq 1,\ i \in \mathcal{V}.$$

Deplano, Franceschelli, and Giua, "Dynamic min and max consensus and size estimation of anonymous multi-agent networks", in *IEEE Trans. Autom. Control* (20223).

DMC Protocol: DYNAMIC MAX-CONSENSUS

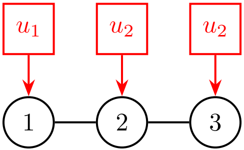
(Input): Tuning parameter $\alpha > 0$
(Initialization): $x_i(0) \in \mathbb{R}$ for $i \in \mathcal{V}$
(Execution): **for** $k = 1, 2, 3, \ldots$ **each node** i **does**

1) Gather $x_j(k-1)$ from each neighbor $j \in \mathcal{N}_i$
2) Update the current state according to

$$x_i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{x_j(k-1) - \alpha, u_i(k)\}$$

Deplano, Franceschelli, and Giua, "Dynamic min and max consensus and size estimation of anonymous multi-agent networks", in *IEEE Trans. Autom. Control* (20223).

A simple example

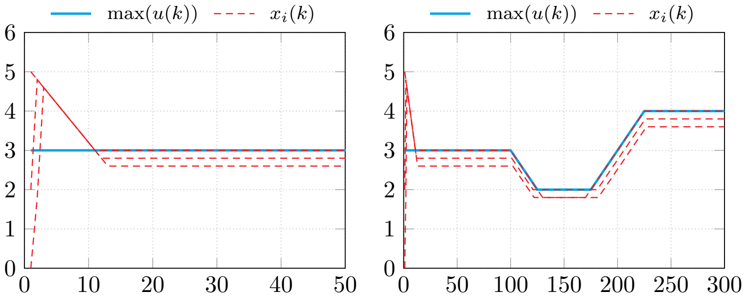


Initial conditions

$u_1(0) = 1,\ x_1(0) = 0$

$u_2(0) = 2,\ x_2(0) = 2$

$u_3(0) = 3,\ x_3(0) = 5$



Remarks:

- Regardless of the initial condition, the agents' state converge to a neighborhood of the maximum value of the inputs that depends on α;
- When the maximum input decreases, the agents' states decreases with a rate given by α.
- Can we do better than this? **Yes**

STDMC Protocol: Self-Tuning Dynamic Max-Consensus

(Input): uning parameters $\alpha^{\text{MAX}} \geq \alpha^{\text{MIN}} > 0$

(Initialization): $x_i(0) \in \mathbb{R}$ for $i \in \mathcal{V}$

$\alpha_i(0) \in \{\alpha^{\text{MIN}}, \alpha^{\text{MAX}}\}$ for $i \in \mathcal{V}$

(Execution): for $k = 1, 2, 3, \ldots$ **each node** i **does**

1) Gather $x_j(k-1)$ and $\alpha_j(k-1)$ from each neighbor $j \in \mathcal{N}_i$
2) Update the current state according to

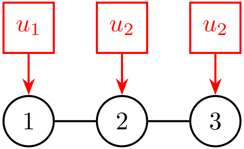
$$x_i(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{x_j(k-1) - \alpha_j(k-1), u_i(k)\}$$

3) Update the current parameter according to

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

Deplano, Franceschelli, and Giua, "Dynamic max–consensus with local self–tuning", in *IFAC–PapersOnLine* (2022).

A simple example



Initial conditions

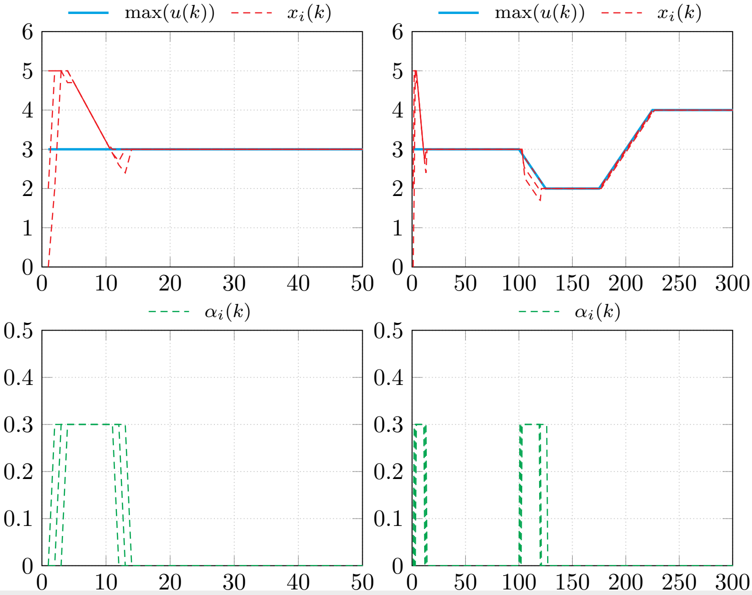
$u_1(0) = 1,\ x_1(0) = 0,\ \alpha_1(0) = 0$

$u_2(0) = 2,\ x_2(0) = 2,\ \alpha_2(0) = 0$

$u_3(0) = 3,\ x_3(0) = 5,\ \alpha_3(0) = 0$

Parameters

$$\alpha^{\text{MIN}} \ll \alpha^{\text{MAX}}$$



Working assumption

Assumption 1

The variation of the reference signals $u_i(k)$ are bounded a constant $\Pi \geq 0$, i.e., for $k \geq 0$ it holds

$$\Delta u_i(k) = |u_i(k) - u_i(k-1)| \leq \Pi$$

Any continuous-time signal with bounded derivative can be over-sampled to reduce its absolute variation.

Working assumption

Assumption 1

The variation of the reference signals $u_i(k)$ are bounded a constant $\Pi \geq 0$, i.e., for $k \geq 0$ it holds

$$\Delta u_i(k) = |u_i(k) - u_i(k-1)| \leq \Pi$$

Any continuous-time signal with bounded derivative can be over-sampled to reduce its absolute variation.

Theorem 1: Tracking error of STDMC Protocol

Consider time-varying reference signals $u_i(k) \in \mathbb{R}$ under Assumption 1 and let $\delta_{\mathcal{G}}$ be the diameter of graph $\mathcal{G}$. If $\mathcal{G}$ is connected and if

$$\alpha^{\text{MAX}} > \Pi,$$

then $\exists T_c \geq 0$ such that the tracking error $e_i(k)$ of each agent is bounded by ε_{tr}, i.e., for $k \geq T_c$ it holds

$$e_i(k) = |x_i(k) - \max(u(k))| \leq \varepsilon_{tr} = (\alpha^{\text{MAX}} + \Pi)\delta_{\mathcal{G}}, \qquad i \in \mathcal{V} \tag{2}$$

and moreover

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

Theorem 2: Steady-state error of STDMC Protocol

If the reference signals remain constant for $k \geq k_0$, then the steady state error $e_i(k)$ of each agent is bounded by ε_{ss}, i.e., for $k \geq k_0 + 2\delta_{\mathcal{G}}$ by ε_{ss},

$$e_i(k) = |x_i(k) - \max(u(k))| \leq \varepsilon_{ss} = \alpha^{\text{MIN}}\delta_{\mathcal{G}}, \qquad i \in \mathcal{V}. \tag{3}$$

Theorem 1: Tracking error of STDMC Protocol

Consider time-varying reference signals $u_i(k) \in \mathbb{R}$ under Assumption 1 and let $\delta_{\mathcal{G}}$ be the diameter of graph $\mathcal{G}$. If $\mathcal{G}$ is connected and if

$$\alpha^{\mathrm{MAX}} > \Pi,$$

then $\exists T_c \geq 0$ such that the tracking error $e_i(k)$ of each agent is bounded by ε_{tr}, i.e., for $k \geq T_c$ it holds

$$e_i(k) = |x_i(k) - \max(u(k))| \leq \varepsilon_{tr} = (\alpha^{\mathrm{MAX}} + \Pi)\delta_{\mathcal{G}}, \qquad i \in \mathcal{V} \tag{2}$$

and moreover

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

Theorem 2: Steady-state error of STDMC Protocol

If the reference signals remain constant for $k \geq k_0$, then the steady state error $e_i(k)$ of each agent is bounded by ε_{ss}, i.e., for $k \geq k_0 + 2\delta_{\mathcal{G}}$ by ε_{ss},

$$e_i(k) = |x_i(k) - \max(u(k))| \leq \varepsilon_{ss} = \alpha^{\mathrm{MIN}}\delta_{\mathcal{G}}, \qquad i \in \mathcal{V}. \tag{3}$$

Theorem 3: Tracking error when the bound Π is unknown

If the agents update their local parameter α_i^{MAX} according to

$$\alpha_i^{\text{MAX}}(k) = \max_{j \in \mathcal{N}_i \cup \{i\}} \{\alpha_j^{\text{MAX}}(k-1), \theta \cdot \Delta u_i(k)\}, \qquad \theta > 1,$$

then $\exists T_c \geq 0$ such that the tracking error $e_i(k)$ is bounded for $k \geq T_c$ by the following

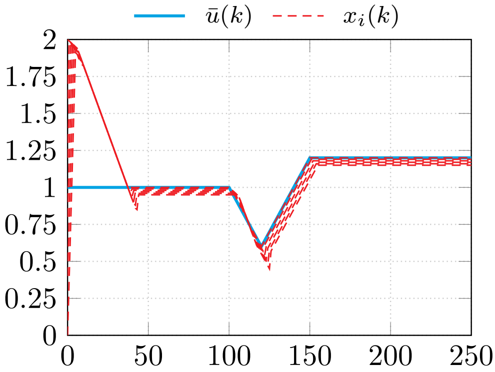
$$e_i(k) = |x_i(k) - \overline{u}(k)| \leq (\theta + 1)\Pi\delta_{\mathcal{G}} = \varepsilon_{tr}, \qquad i \in \mathcal{V}. \tag{4}$$

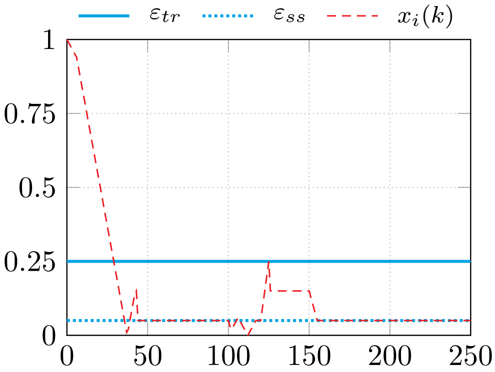
Main features of the STDMC Protocol

- **Robustness** to re-initialization;
- **Scalability** in large networks: both the memory burden and the number of exchanged messages do not increase with the dimension of the network.
- **Self-tuning logic** to achieve arbitrary small steady state error and to cope with unknown inputs variation;
- **Boundedness** of the tracking error.

Piece-wise linear inputs - Line graph of 6 nodes

$\alpha^{\text{MAX}} = 0.03$ $\alpha^{\text{MIN}} = 0.01$ $\Pi = 0.02$ $\varepsilon_{tr} = 0.25$ $\varepsilon_{ss} = 0.05$

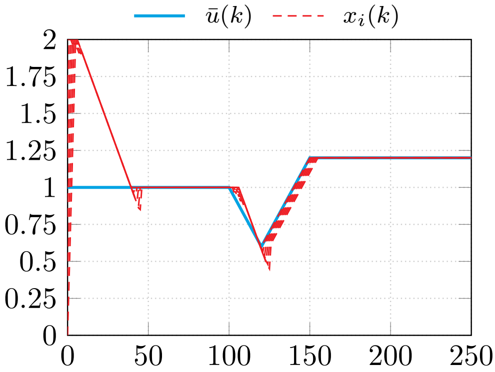


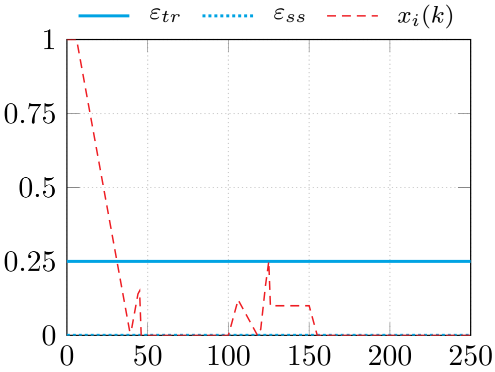


Piece-wise linear inputs - Line graph of 6 nodes

$\alpha^{\mathrm{MAX}} = 0.03$ $\alpha^{\mathrm{MIN}} = 1 \cdot 10^{-8}$ $\Pi = 0.02$ $\varepsilon_{tr} = 0.25$ $\varepsilon_{ss} = 5 \cdot 10^{-8}$

Arbitrary small steady state error!

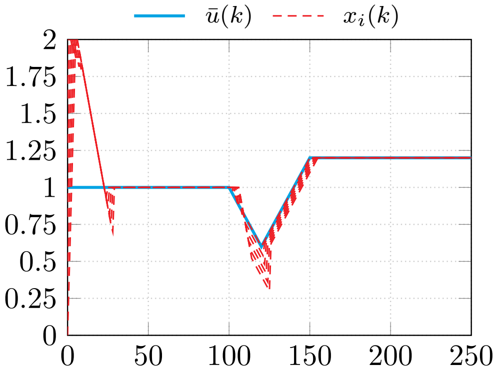


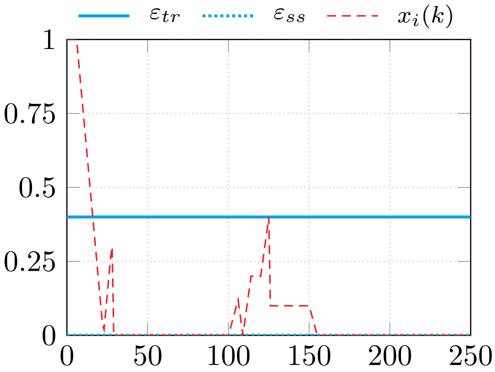


Piece-wise linear inputs - Line graph of 6 nodes

$\alpha^{\text{MAX}} = 0.06$ $\alpha^{\text{MIN}} = 1 \cdot 10^{-8}$ $\Pi = 0.02$ $\varepsilon_{tr} = 0.40$ $\varepsilon_{ss} = 5 \cdot 10^{-8}$

Faster convergence rate! But higher tracking error...

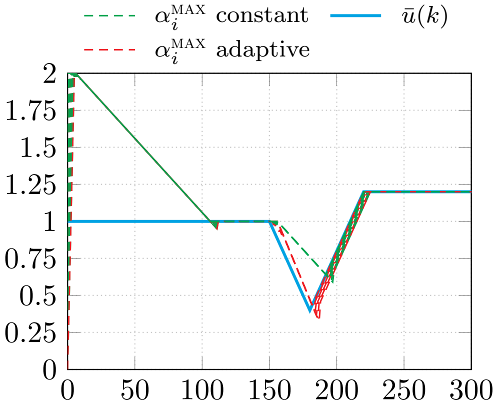


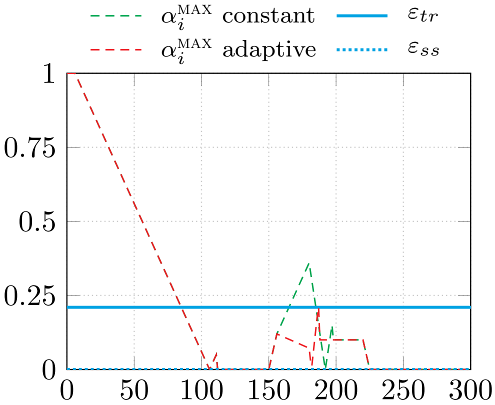


Unknown bound Π - Line graph of 6 nodes

$\alpha_i^{\text{MAX}}(0) = 0.01$ $\alpha^{\text{MIN}} = 1 \cdot 10^{-8}$ $\Pi = 0.02$ $\theta = 1.1$ $\varepsilon_{tr} = 0.22$ $\varepsilon_{ss} = 5 \cdot 10^{-8}$

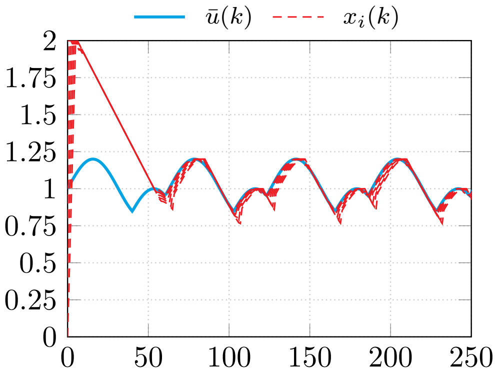
The assumption $\alpha_i^{\text{MAX}}(0) > \Pi$ is not satisfied!

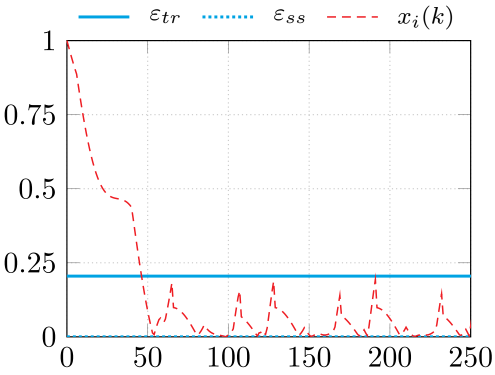




Sinusoidal inputs - Line graph of 6 nodes

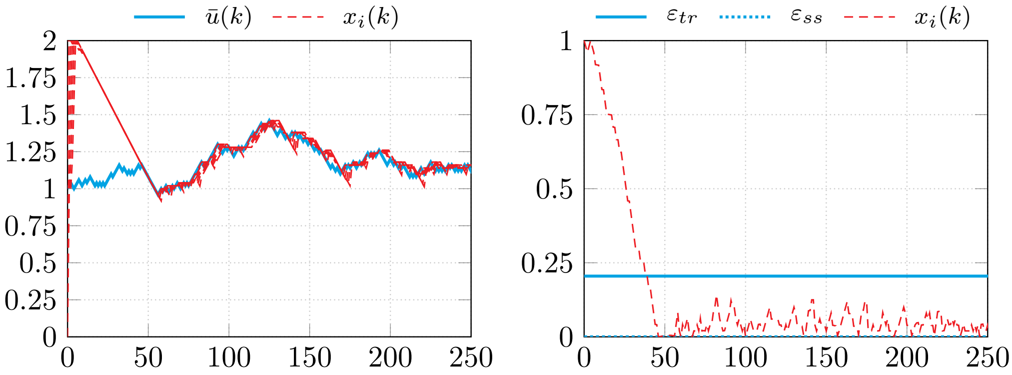
$\alpha^{\mathrm{MAX}} = 0.06$ $\alpha^{\mathrm{MIN}} = 1 \cdot 10^{-8}$ $\Pi = 0.02$ $\varepsilon_{tr} = 0.40$ $\varepsilon_{ss} = 5 \cdot 10^{-8}$





Random inputs - Random graph of 100 nodes with diameter $\delta_{\mathcal{G}} = 5$

$\alpha^{\text{MAX}} = 0.06$ $\alpha^{\text{MIN}} = 1 \cdot 10^{-8}$ $\Pi = 0.02$ $\varepsilon_{tr} = 0.40$ $\varepsilon_{ss} = 5 \cdot 10^{-8}$



Outline

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

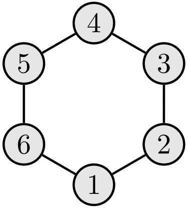
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Initial time $k = 0$



$$e_i(0) = 3, \quad \forall i \in V$$

$$d(0) = r(0) = 3$$

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

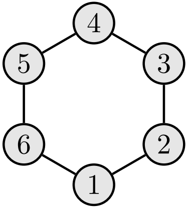
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Initial time $k = 0$



$$e_i(0) = 3, \quad \forall i \in V$$

$$d(0) = r(0) = 3$$

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

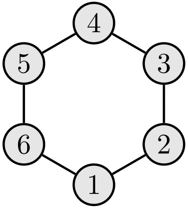
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Initial time $k = 0$



$$e_i(0) = 3, \quad \forall i \in V$$

$$d(0) = r(0) = 3$$

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

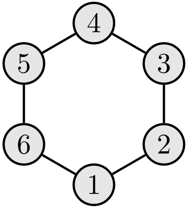
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Initial time $k = 0$



$$e_i(0) = 3, \quad \forall i \in V$$

$$d(0) = r(0) = 3$$

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

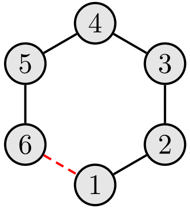
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Time of change $k = k^*$



$$e_i(k^*) = \cdots, \quad \forall i \in V$$

$$d(k^*) = \cdots, \quad r(k^*) = \cdots$$

Graph parameters of interest

Let $n(k)$ be the time-varying number of agents and $\text{dist}_{ij}(k)$ be the length of the shortest path between agents i, j, then:

- The eccentricity $e_i(k)$ of node $i \in V$ at time k is defined as the maximal distance from i of any other node,

$$e_i(k) = \max_{j \in V} \text{dist}_{ij}(k).$$

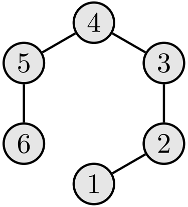
- The diameter $d(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the maximal eccentricity among the nodes,

$$d(k) = \max_{i \in V} e_i(k).$$

- The radius $r(k)$ of graph $\mathcal{G}(k)$ at time k is defined as the minimal eccentricity among the nodes,

$$r(k) = \min_{i \in V} e_i(k).$$

Time of change $k = k^*$



$$e_1(k^*) = e_6(k^*) = 5$$

$$e_2(k^*) = e_5(k^*) = 4$$

$$e_3(k^*) = e_4(k^*) = 3$$

$$d(k^*) = 5, \qquad r(k^*) = 3$$

Cardinality estimation: methodology

The methodology is based on statistical inference concepts and can be outlined in three steps:

1. **Generation**: When a node i joins the network, it generates $p > 0$ independent random numbers $u_{ij} \in [0,1]$ from a uniform distribution, i.e., $u_{ij} \sim U(0,1)$ with $j = 1, \ldots, p$;
2. **Estimation**: The $n(k)$ active nodes execute the STDMC Protocol, thus each node i computes p estimates x_{ij} with $j = 1, \ldots, p$ of the maximum value among each local set $[u_{1j}, u_{2j}, \ldots]$;
3. **Inference**: Each node i infers the estimate $\hat{n}_i(k)$ of the network size $n(k)$ by maximum likelihood estimation from its own set of estimations $[x_{i1}, \ldots, x_{ip}]$.

Varagnolo, Pillonetto, and Schenato, "Distributed cardinality estimation in anonymous networks", in *IEEE Trans. Autom. Control* (2014)

Diameter and radius estimation: methodology

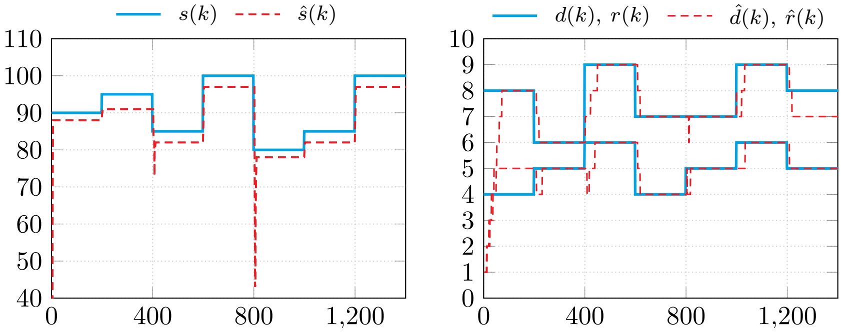
The methodology exploits the specific steady state error of the STDMC protocol and can be outlined in three steps:

1. **Generation**: When a node i joins the network, it generates $p > 0$ independent random numbers $u_{ij} \in [0, 1]$ from a uniform distribution, i.e., $u_{ij} \sim U(0, 1)$ with $j = 1, ..., p$;
2. **Estimation**: The $n(k)$ active nodes execute two instances of the STDMC Protocol with different parameters α^{MIN} to estimate the maximum value among each local set $[u_{1j}, u_{2j}, ...]$:
   - Let x_{ij} be the estimates when $\alpha^{\mathrm{MIN}} = \alpha^*$ for $j = 1, ..., p$;
   - Let y_{ij} be the estimates when $\alpha^{\mathrm{MIN}} = \beta^*$ for $j = 1, ..., p$.
3. **Inference**: Each node i computes p estimates ε_{ij} with $j = 1, \ldots, p$ of their own eccentricity by exploiting the different steady state errors between x_{ij} and y_{ij} due to $\beta^* > \alpha^* > 0$. Then, it infers the diameter and the radius by running another instance of the STDMC Protocol to estimate z_{ij} the maximum value among each local set of variables $[\varepsilon_{1j}, \varepsilon_{2j}, \ldots]$.

Deplano, Franceschelli, and Giua, "Distributed tracking of graph parameters in anonymous networks with time–varying topology", in *IEEE Conference on Decision and Control*, (2021)

Simulation with a random graph of $n(k) \in [80, 100]$ nodes

$\alpha^{\text{MAX}} = 0.1$, $\alpha^{\text{MIN}} = 10^{-12}$, cardinality $s(k)$, diameter $d(k)$, radius $r(k)$



Outline

Consensus via distributed optimization

The dynamic consensus problems on average, maximum, and median functions can be recast as distributed time-varying optimization problems of the following type

$$\begin{aligned} x^*(k) = \underset{x_1,\ldots,x_n}{\operatorname{argmin}} \; & \sum_{i=1}^{n} \frac{1}{p} |x_i - u_i(k)|^p \\ \text{s.t.} \quad & x_i = x_j \qquad \forall\, (i,j) \in \mathcal{E} \\ & x_i \in \mathcal{X}_{i,k} \quad\; \forall\, i \in \mathcal{V}. \end{aligned}$$

where:

i) If $p = 2$ and $\mathcal{X}_{i,k} = \mathbb{R}$, then $x^*(k) = \operatorname{avg}(u(k))$;

ii) If $p = 2$ and $\mathcal{X}_{i,k} = \{x \geq u_i(k)\}$, then $x^*(k) = \max(u(k))$;

iii) If $p = 1$ and $\mathcal{X}_{i,k} = \mathbb{R}$, then $x^*(k) = \operatorname{med}(u(k))$.

Reference work: Deplano, D., Bastianello, N., Franceschelli, M., & Johansson, K. H. (2023, December), "A unified approach to solve the dynamic consensus on the average, maximum, and median values with linear convergence", in: *62nd IEEE Conference on Decision and Control (CDC)*, (2023)

ADMM-based protocols

Dynamic average consensus protocol:

$$x_i(k) = \frac{u_i(k) + \sum_{j \in \mathcal{N}_i} z_{ij}(k-1)}{1 + \rho\eta_i};$$

Dynamic maximum consensus protocol:

$$x_i(k) = \max\left\{u_i(k), \frac{u_i(k) + \sum_{j \in \mathcal{N}_i} z_{ij}(k-1)}{1 + \rho\eta_i}\right\};$$

Dynamic median consensus protocol (Eq. (5)):

$$x_i(k) = u_i(k) + \max\{\theta_i^- - u_i(k), 0\} + \min\{\theta_i^+ - u_i(k), 0\}, \qquad \theta_i^\pm(k) = \frac{\sum_{j \in \mathcal{N}_i} z_{ij}(k-1) \pm 1}{\rho\eta_i};$$

where the auxiliary variables are updated according to

$$z_{ij}(k) = (1-\alpha)z_{ij}(k-1) + \alpha(2\rho x_j(k) - z_{ji}(k-1))$$

Theorem 4: Tracking error of ADMM-based protocols

Consider time-varying reference signals $u_i(k) \in \mathbb{R}$ under Assumption 1. The tracking error converges linearly to a neighborhood of the consensus with a bounded radius proportional to Π and $\sqrt{|\mathcal{E}|}$.

Corollary: Steady-state error of of ADMM-based protocols

If the reference signals remain constant, then the steady state error is zero.

Remarks:

- The main novelty is the linear convergence, indeed, ADMM-based protocol are known to converge with sub-linear rate for convex problems that are not strongly convex;
- An advantage of this approach is that the agents do not need to know (or compute) the inputs' derivatives;
- A disadvantage of this approach is that each agent needs to store and share a number of variables which is the double the number of neighbors;
- The convergence occurs asymptotically and not in finite-time.

Theorem 4: Tracking error of ADMM-based protocols

Consider time-varying reference signals $u_i(k) \in \mathbb{R}$ under Assumption 1. The tracking error converges linearly to a neighborhood of the consensus with a bounded radius proportional to Π and $\sqrt{|\mathcal{E}|}$.

Corollary: Steady-state error of of ADMM-based protocols

If the reference signals remain constant, then the steady state error is zero.

Remarks:

- The main novelty is the linear convergence, indeed, ADMM-based protocol are known to converge with sub-linear rate for convex problems that are not strongly convex;
- An advantage of this approach is that the agents do not need to know (or compute) the inputs' derivatives;
- A disadvantage of this approach is that each agent needs to store and share a number of variables which is the double the number of neighbors;
- The convergence occurs asymptotically and not in finite-time.

Theorem 4: Tracking error of ADMM-based protocols

Consider time-varying reference signals $u_i(k) \in \mathbb{R}$ under Assumption 1. The tracking error converges linearly to a neighborhood of the consensus with a bounded radius proportional to Π and $\sqrt{|\mathcal{E}|}$.

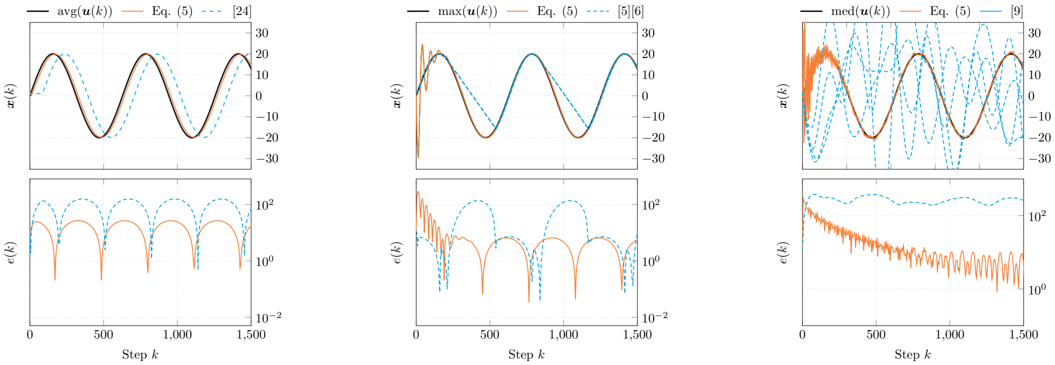
Corollary: Steady-state error of of ADMM-based protocols

If the reference signals remain constant, then the steady state error is zero.

Remarks:

- The main novelty is the linear convergence, indeed, ADMM-based protocol are known to converge with sub-linear rate for convex problems that are not strongly convex;
- An advantage of this approach is that the agents do not need to know (or compute) the inputs' derivatives;
- A disadvantage of this approach is that each agent needs to store and share a number of variables which is the double the number of neighbors;
- The convergence occurs asymptotically and not in finite-time.

Simulations in a network with $n \approx 100$ nodes



[5] Deplano, Franceschelli, and Giua, "Dynamic max–consensus with local self–tuning", in *IFAC–PapersOnLine* (2022)
[9] Vasiljevic, Petrovic, Arbanas, and Bogdan, "Dynamic median consensus for marine multi–robot systems using acoustic communication", in *IEEE Robotics and Automation Letters* (2020)

[24] Franceschelli and Gasparri, "Multi-stage discrete time and randomized dynamic average consensus", in *Automatica* (2019)

Outline

Future directions

Possible extensions:

- How to deal with asynchronous activation of the agents and packet losses?
- How do noise in the measurements and delays in the communications affect the tracking?
- How to make the protocol robust against malicious agents?

Possible applications:

- Optimal reactive power allocation for distributed renewable energy sources;
- Real-time distributed monitoring in wide-area systems for long-term voltage instability;
- ...

Future directions

Possible extensions:

- How to deal with asynchronous activation of the agents and packet losses?
- How do noise in the measurements and delays in the communications affect the tracking?
- How to make the protocol robust against malicious agents?

Possible applications:

- Optimal reactive power allocation for distributed renewable energy sources;
- Real-time distributed monitoring in wide-area systems for long-term voltage instability;
- ...

Dynamic Max-consensus in Multi-Agent systems and Graph parameter Estimation in Open Networks

Thank you for your attention!

Diego Deplano

Email: diego.deplano@unica.it
Webpage: https://sites.google.com/view/deplanodiego/