Improved Consensus Algorithms With Memory

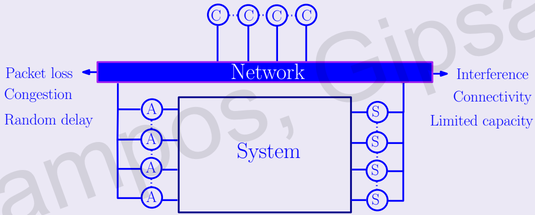
Gabriel Rodrigues de Campos
Alexandre Seuret

NeCS Team
CNRS - GIPSA-Lab Automatic Control Department
INRIA Rhône-Alpes

50^{th} IEEE Conference on Decision and Control and European Control Conference

Networked Control Systems (NCS)

- Systems wherein the control loops are closed through a real-time network.

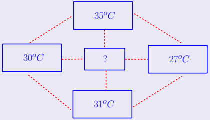


Defining feature

Control and feedback signals are exchanged among the system's components in the form of information packages through a network.

Agreement Algorithms

Consensus

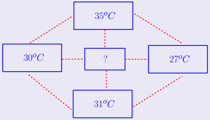


Common Value

- Olfati-Saber et al., IEEE Trans. Autom. Control [2004]
- Ren et al., IEEE CDC [2008]

Agreement Algorithms

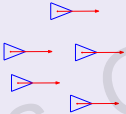
Consensus



Common Value

- Olfati-Saber et al., IEEE Trans. Autom. Control [2004]
- Ren et al., IEEE CDC [2008]

Flocking

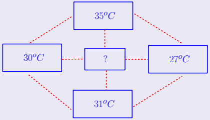


Common Velocity

- Olfati-Saber, IEEE Trans. Autom. Control [2006]
- Cortes et al., IEEE Trans. Robot. Autom. [2004]

Agreement Algorithms

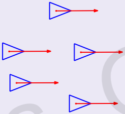
Consensus



Common Value

- Olfati-Saber et al., IEEE Trans. Autom. Control [2004]
- Ren et al., IEEE CDC [2008]

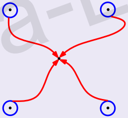
Flocking



Common Velocity

- Olfati-Saber, IEEE Trans. Autom. Control [2006]
- Cortes et al., IEEE Trans. Robot. Autom. [2004]

Rendez-vous

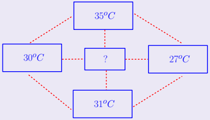


Common Position

- Lin et al., IEEE CDC [2003]
- Cortes et al., IEEE Trans. Robot. Autom. [2004]

Agreement Algorithms

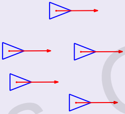
Consensus



Common Value

- Olfati-Saber et al., IEEE Trans. Autom. Control [2004]
- Ren et al., IEEE CDC [2008]

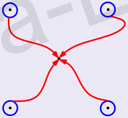
Flocking



Common Velocity

- Olfati-Saber, IEEE Trans. Autom. Control [2006]
- Cortes et al., IEEE Trans. Robot. Autom. [2004]

Rendez-vous



Common Position

- Lin et al., IEEE CDC [2003]
- Cortes et al., IEEE Trans. Robot. Autom. [2004]

Question

Is it possible to improve the convergence rate? And how?

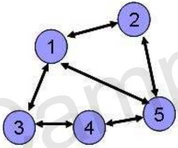
Content

- Problem Statement
- Model definition
- Improved behavior guarantees
- Stability analysis
- Examples
- Conclusions

Consensus in multi-agent systems (MAS):

To reach an agreement regarding a certain quantity of interest that depends on the state of all agents under limited communication

Applications: multi-robot systems, distributed estimation and filtering in networked systems.



$$\Longleftrightarrow \begin{cases} \dot{x}_i = u_i \\ u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \end{cases} \Longleftrightarrow \quad \dot{x} = -Lx$$

$$\boxed{x(\infty) = \frac{\sum_i^5 x_i(0)}{5}\vec{1}}$$

$$L = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & -1 & 0 & -1 & 3 \end{bmatrix}$$

where $L = \Delta - A$ is the Laplacian matrix, and A is the adjacency matrix.

Simple Integrator Consensus

Consider the classical simple integrator consensus algorithm

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \quad i \in \{1, \ldots, N\}, \tag{1}$$

or, expressed in another way,

$$\dot{x}(t) = -Lx(t) \,, \tag{2}$$

where x represents the vector containing the agents variables.

Simple Integrator Consensus

Consider the classical simple integrator consensus algorithm

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \quad i \in \{1, \ldots, N\}, \qquad (1)$$

or, expressed in another way,

$$\dot{x}(t) = -Lx(t) \, , \qquad (2)$$

where x represents the vector containing the agents variables.

Remark:

Convergence rate is related with the 2^{nd} smallest eigenvalue of L, λ_2

Simple Integrator Consensus

Consider the classical simple integrator consensus algorithm

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \quad i \in \{1, \ldots, N\}, \tag{1}$$

or, expressed in another way,

$$\dot{x}(t) = -Lx(t) \ , \tag{2}$$

where x represents the vector containing the agents variables.

Main Idea: Stabilizing Delay (Michiels et al.[2004])

Introduce *local memory* in the algorithm to improve convergence performances

Simple Integrator Consensus

Consider the classical simple integrator consensus algorithm

$$\begin{cases} \dot{x}_i(t) = u_i(t) \\ u_i(t) = \sum_{j \in \mathcal{N}_i} a_{ij}(x_j(t) - x_i(t)) \end{cases} \quad i \in \{1, \ldots, N\}, \tag{1}$$

or, expressed in another way,

$$\dot{x}(t) = -Lx(t) \,, \tag{2}$$

where x represents the vector containing the agents variables.

As previously in

Rodrigues de Campos et al., IFAC NecSys [2010]
"Continuous-time double integrator consensus algorithms improved by an appropriate sampling"

Delayed Consensus Algorithms

The previous algorithm is modified into a new algorithm defined by

$$\dot{x}(t) = -(L + \delta A)x(t) + \delta A x(t - \tau) \tag{3}$$

Note that if δ and/or τ are taken as zeros, then the classical algorithm is retrieved.

Remark:

Algorithm's convergence proprieties remain intact

Delayed Consensus Algorithms

The previous algorithm is modified into a new algorithm defined by

$$\dot{x}(t) = -(L + \delta A)x(t) + \delta A x(t - \tau) \quad (3)$$

Note that if δ and/or τ are taken as zeros, then the classical algorithm is retrieved.

Remark:

Algorithm's convergence proprieties remain intact

Drawback:

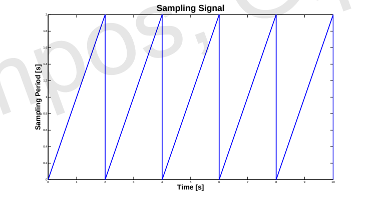
- Large memory is needed in order to store all x values over the whole time window $[t - \tau, t]$

Sampling Delay (Fridman et al. [2004])

We will consider a sampling delay such that:

$$\tau(t) = t - t_k,\ t_k \leq t < t_{k+1}\ ,$$

where the t_k's corresponds to the sampling instants and $T = t_{k+1} - t_k$ is the sampling period.



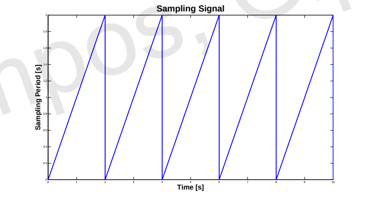
Advantages: Smaller memory requirement

Sampling Delay (Fridman et al. [2004])

We will consider a sampling delay such that:

$$\tau(t) = t - t_k,\ t_k \leq t < t_{k+1}\ ,$$

where the t_k's corresponds to the sampling instants and $T = t_{k+1} - t_k$ is the sampling period.

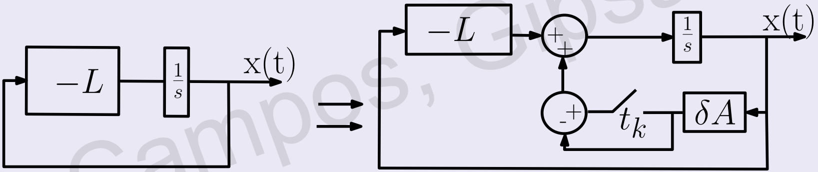


Advantages: Smaller memory requirement
Drawbacks: More dedicated stability analysis

Finally the proposed algorithm is

$$\forall t \in [t_k \; t_{k+1}[, \quad \dot{x}(t) = (-L - \delta A)x(t) + \delta A x(t_k) \tag{4}$$

where δ and T are now two additional control parameters.



Bloc diagrams of the classical and the improved algorithms

Control Strategy:

Continus time system + ”sampled and held” data as *local memory*

Consensus agreement:

Proof that the proposed algorithm achieve consensus

Consensus agreement:

Proof that the proposed algorithm achieve consensus

Considering a performance optimisation:

Proposal of a method to choose appropriately the algorithm parameters δ and T for a given L

Consensus agreement:

Proof that the proposed algorithm achieve consensus

Considering a performance optimisation:

Proposal of a method to choose appropriately the algorithm parameters δ and T for a given L

Stability:

Establishment of exponential stability conditions.

Definition: Let $\alpha > 0$ be some positive, constant, real number. The system is said to be exponentially stable with the decay rate α, or α-stable, if there exists a scalar $\beta \geq 1$ such that the solution $x(t; t_0, \phi)$ satisfies:

$$|x(t; t_0, \phi)| \leq \beta |\phi|_\tau e^{-\alpha(t-t_0)}.$$

Assumptions on the multi-agent set:

- A1. Communication graph with a directed spanning tree
- A2. Sampling process is periodic
- A3. All agents are synchronized and share the same clock
- A4. $L = \mu I - A$ (Not restrictive)

Assumptions on the multi-agent set:

- A1. Communication graph with a directed spanning tree
- A2. Sampling process is periodic
- A3. All agents are synchronized and share the same clock
- A4. $L = \mu I - A$ (Not restrictive)

Problems to be solved:

- P1. Theoretical guarantee of improved behavior
- P2. Analytic expression of the consensus point
- P3. Convergence to this point
- P4. Convergence rate to this point

Model Transformation

For sake of generalization, let μ be a positive scalar such that:

$$\sum_{j\in\mathcal{N}_i} a_{ij} = \mu, \quad i \in \{1,\ldots,N\}.$$

Is then possible to make a change of coordinates $x = Wz$ such that

$$ULW = \begin{bmatrix} B & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}, \tag{5}$$

where $B \in \mathbb{R}^x$, and for graphs containing a directed spanning tree, $U = \begin{bmatrix} U_1^T & U_2^T \end{bmatrix}^T = W^{-1}$ and $U_2 = (U)_N$ corresponds to the N^{th} line of U.

The consensus problem (4) can be rewritten using $z_1 \in \mathbb{R}^{N-1}$, $z_2 \in \mathbb{R}$ and the matrix B is given in (8):

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k), \tag{6a}$$

$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k), \tag{6b}$$

Model Transformation

The consensus problem (4) can be rewritten using $z_1 \in \mathbb{R}^{N-1}$, $z_2 \in \mathbb{R}$ and the matrix B is given in (8):

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k), \quad (7a)$$
$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k), \quad (7b)$$

Interpretation:

The sampled algorithm is decomposed into two components:

- z_1 is a vectorial component associated to non-zero eigenvalues that converge to zero.

Model Transformation

The consensus problem (4) can be rewritten using $z_1 \in \mathbb{R}^{N-1}$, $z_2 \in \mathbb{R}$ and the matrix B is given in (8):

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k), \qquad (7a)$$

$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k), \qquad (7b)$$

Interpretation:

The sampled algorithm is decomposed into two components:

- z_1 is a vectorial component associated to non-zero eigenvalues that converge to zero.
- z_2 is scalar component associated to the zero eigenvalue that converge to the initial positions average.

Theoretical guarantee for improved behavior

Proposition: For small values of δ and T convergence increases when compare with the trivial algorithm.

Let B be the diagonal matrix of the Laplacian matrix eigenvalues such that

$$B = \begin{bmatrix} -\lambda_2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -\lambda_N \end{bmatrix}. \quad (8)$$

Theoretical guarantee for improved behavior

Proposition: For small values of δ and T convergence increases when compare with the trivial algorithm.

Let B be the diagonal matrix of the Laplacian matrix eigenvalues such that

$$B = \begin{bmatrix} -\lambda_2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & -\lambda_N \end{bmatrix}. \tag{8}$$

Thus, we establish for all $i = 1, \dots, N-1$

$$\dot{z}_{1i}(t) = (-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu))z_{1i}(t) - \delta(\lambda_{i+1} + \mu)z_{1i}(t_k). \tag{9}$$

Theoretical guarantee for improved behavior

By integrating the previous equation, the following recurrence equation represents the discrete dynamics of the algorithm.

$$z_{1i}(t_{k+1}) = A(\lambda_{i+1}, \delta, T) z_{1i}(t_k), \tag{10}$$

with

$$A(\lambda_{i+1}, \delta, T) = \exp^{(-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu))T} \frac{-\lambda_{i+1}}{-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu)} + \frac{\delta(\lambda_{i+1} + \mu)}{-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu)}.$$

Theoretical guarantee for improved behavior

By integrating the previous equation, the following recurrence equation represents the discrete dynamics of the algorithm.

$$z_{1i}(t_{k+1}) = A(\lambda_{i+1}, \delta, T) z_{1i}(t_k), \tag{10}$$

with

$$A(\lambda_{i+1}, \delta, T) = \exp^{(-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu))T} \frac{-\lambda_{i+1}}{-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu)} + \frac{\delta(\lambda_{i+1} + \mu)}{-\lambda_{i+1} + \delta(\lambda_{i+1} + \mu)}.$$

We will show that by varying δ and T values close to zero, we achieve a performance improvement for $\forall \lambda_{i+1}$, if

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} \leq 0, \text{ for some } \delta \text{ values} \tag{11a}$$

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} \leq 0, \text{ for some T values} \tag{11b}$$

Theoretical guarantee for improved behavior

When we evaluate the previous equation for $T \simeq 0$ and for $\delta \simeq 0$, respectively, we have

$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial T} = -\lambda_{i+1} \leq 0$$
$$\frac{\partial A(\lambda_{i+1}, \delta, T)}{\partial \delta} = e^{-\lambda_{i+1} T}(\lambda_{i+1} + \mu)\left(T + \frac{1}{\lambda_{i+1}}\right) - \left(\frac{\lambda_{i+1} + \mu}{\lambda_{i+1}}\right) \leq 0$$

Theoretical guarantee for improved behavior

When we evaluate the previous equation for $T \simeq 0$ and for $\delta \simeq 0$, respectively, we have

$$\frac{\partial A(\lambda_{i+1},\delta,T)}{\partial T} = -\lambda_{i+1} \leq 0$$

$$\frac{\partial A(\lambda_{i+1},\delta,T)}{\partial \delta} = e^{-\lambda_{i+1}T}(\lambda_{i+1}+\mu)\left(T+\frac{1}{\lambda_{i+1}}\right) - \left(\frac{\lambda_{i+1}+\mu}{\lambda_{i+1}}\right) \leq 0$$

Conclusion:

As $\frac{\partial A(\lambda_{i+1},\delta,T)}{\partial T} = -\lambda_{i+1}$ is negative for all value of δ, and $\frac{\partial A(\lambda_{i+1},\delta,T)}{\partial \delta}$ is also negative for small values of T, we can then conclude that for small values of δ and T we converge more rapidly when compare with the trivial algorithm.

Sketch of the Proof:

Step 1)

$$\dot{x}(t) = (-L - \delta A)x(t) + \delta A x(t_k)$$

$\Downarrow$ *Model Transformation*

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k),$$
$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k),$$

Sketch of the Proof:

Step 2) Stability of z_2

For $\forall t \in [t_k \ t_{k+1}[$

$$z_2(t)(= U_2 x(t)) = z_2(t_k) = z_2(0)$$

proving that z_2 is constant and that

$$x(\infty) = U_2 x(0)$$

Sketch of the Proof:

Step 3) Exponential stability of z_1

Consider the following Functional:

$$\bar{V}(t, z_1(t)) = z_1^T(t) P z_1(t)$$

The objective is to prove that the increment ΔV_α is negative definite:

$$\Delta V_\alpha = \bar{V}(k+1) - e^{-2\alpha T} \bar{V}(k) < 0,$$

then $z_1(t) \rightarrow_{t \rightarrow \infty} 0$ (with a exp. decay rate α)

Main Result

Consider now the following dynamics of z_1

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k), \quad (13a)$$

$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k), \quad (13b)$$

and re-write it in the following way

$$\dot{z}_1(t) = A(\delta)z_1(t) + A_d(\delta)z_1(t_k),$$

with $A(\delta) = (-B + \delta(B + \mu I))$ and $A_d(\delta) = -\delta(B + \mu I)$.

Main Result (based on A. Seuret,"Stability of sampled-data Systems", Automatica [2011])

Assume that there exist $P > 0$, $R > 0$ and S_1 and $X \in \mathbb{S}^n$ and two matrices $S_2 \in \mathbb{R}^{n\times n}$ and $N \in \mathbb{R}^{2n\times n}$ that satisfy

$$\Pi_1 + f_\alpha(T,0)\Pi_2 + h_\alpha(T,0)\Pi_3 < 0, \tag{14}$$

$$\begin{bmatrix} \Pi_1 + h_\alpha(T,T)\Pi_3 & g_\alpha(T,T)N \\ * & -g_\alpha(T,T)R \end{bmatrix} < 0, \tag{15}$$

where

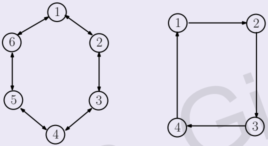
$$\begin{array}{ll} f_\alpha(T,\tau) = (e^{2\alpha(T-\tau)} - 1)/2\alpha, & \Pi_1 = 2He\{M_1^T P(M_0 + \alpha M_1)\} - M_3^T S_1 M_3 \\ g_\alpha(T,\tau) = e^{2\alpha T}(1 - e^{-2\alpha\tau})/2\alpha, & \quad -2He\{M_3^T S_2 M_2) - 2He\{NM_3\}, \\ h_\alpha(T,\tau) = \frac{1}{\alpha}\left[\frac{e^{2\alpha T}-1}{2\alpha T} - e^{2\alpha\tau}\right], & \Pi_2 = M_0^T R M_0 + 2He\{M_0^T(S_1M_3 + S_2M_2)\}, \\ & \Pi_3 = M_2^T X M_2. \end{array} \tag{16}$$

Also,
$M_0 = [\ A(\delta)\quad A_d(\delta)\ ]$, $M_1 = [\ I\quad 0\ ]$, $M_2 = [\ 0\quad I\ ]$, $M_3 = [\ I\quad -I\ ]$, and $2He\{A\} = A + A^T$.

Then, the consensus algorithm is thus $\alpha-$stable and converge to the average of initial conditions, and $\Delta V_\alpha < 0$

Simulation Scenario

Consider a set of agents connected through the undirected and directed graphs shown as in:



Graph 0 Graph 1

To each graph is associated a Laplacian matrix given by

$$L_0 = \begin{bmatrix} 1 & -0.5 & 0 & 0 & 0 & -0.5 \\ -0.5 & 1 & -0.5 & 0 & 0 & 0 \\ 0 & -0.5 & 1 & -0.5 & 0 & 0 \\ 0 & 0 & -0.5 & 1 & -0.5 & 0 \\ 0 & 0 & 0 & -0.5 & 1 & -0.5 \\ -0.5 & 0 & 0 & 0 & -0.5 & 1 \end{bmatrix}, \; L_1 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{bmatrix},$$

and for simulations we took as initial conditions:

$$x_0^T(0) = [30\ 25\ 15\ 0\ -10\ -30] \text{ and } x_1^T(0) = [30\ 25\ 15\ 0].$$

Controller parameters optimization results

$$\max \alpha(\delta, T)$$

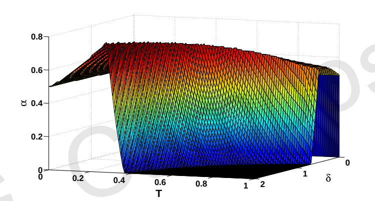


Figure: Exponential decay rate for G_0

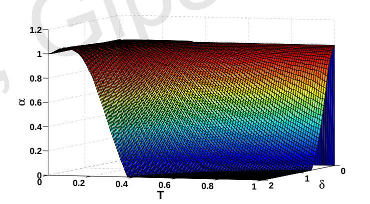
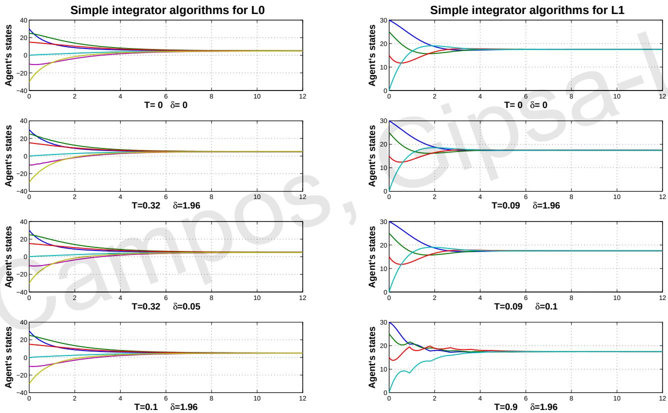


Figure: Exponential decay rate for G_1

Algorithm Convergence

Evolution of the agents state for several values of (δ, T)



Algorithm Convergence (Error with respect to the agreement value evolution)

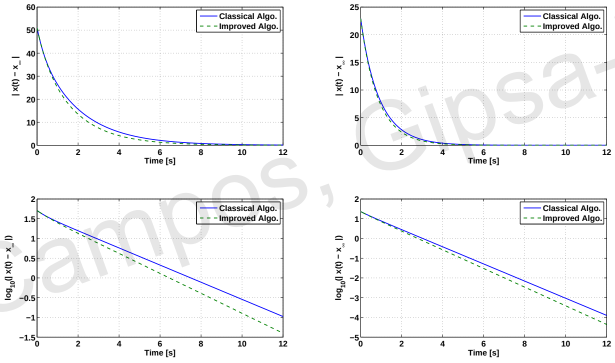
Error with respect to the agreement value evolution:

Consider now

$$\varepsilon = |x(t) - x_\infty|,$$

as the module of the error between agents states and the agreement value x_∞

Algorithm Convergence (Evolution of the error with respect to the agreement value)



Time evolution of error ε and $log_{10}(\varepsilon)$ for G_0 and G_1

Conclusions and Perspectives

For the proposed algorithm

- Theoretical guarantee for improved behavior is stated
- Sufficient stability conditions are provided
- Exponential stability of the solutions is achieved
- Improved behavior observed for different types of networks

Drawbacks

- LMI based stability conditions complexity for large networks.
- Centralized LMI solution

Perspectives

- Robustness with respect to errors in the synchronisation clocks.

Thank you for your attention