Improved Consensus Algorithms using Memory Effects

Gabriel Rodrigues de Campos, Alexandre Seuret

Improved Consensus Algorithms using Memory Effects

Gabriel Rodrigues de Campos, Alexandre Seuret,
NeCS Team, Automatic Control Department of GIPSA-Lab,
UMR CNRS 5216, Grenoble, France
E-mail: gabriel.rodrigues-de-campos,alexandre.seuret@gipsa-lab.grenoble-inp.fr

***Abstract*— This paper deals with simple integrator consensus problems. It is well-known that introducing a delay leads in general to a reduction of performances or to instability. Therefore, investigating the effect of time-delays in consensus problems is an important issue. The objective is the design of a improved consensus algorithm for continuous-time multi-agent systems. The novel algorithm proposes to sampled, in an appropriate manner, part of the multi-agent systems information such that the algorithm converges, assuming that at each instant, agent's control laws will also consider the sampled past information of its neighbors. Stability conditions expressed in terms of LMI's and based on algebraic communication matrix structure are provided. The efficiency of the method is tested for different network communication schemes.**

I. Introduction

Network control systems (NCS) are spatially distributed systems with a communication network used between sensors, actuators, and identifies control over networks as one of the key future directions for control. Their primary advantages includes their low cost, reduced weight, and power requirements, simple installation and maintenance, and higher reliability.

This means NCS's applications can be found in a large range of areas such as mobile sensor networks ([1]), remote surgery, haptic collaboration over Internet, multi-robot systems ([2]), automated highway systems, averaging in communication networks ([3]) and formation control ([4]). Several results have appeared in recent literature that consider systems with different motion models, symmetry of communication and network interactions. A recent review of the vast literature in the field can be found in [5], [6] and [7].

We consider a "consensus" algorithm (or protocol) as an interaction rule that specifies the information exchange between an agent and all of its neighbors over the network in order to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. However, the use of a shared network introduces new challenges, such as delays over communications, packet losses or even communication blackout, witch can dramatically affect "consensus" convergence rate and cooperative control laws efficiency ([8]). Here, we consider that agents are assumed to obey a simple integrator model.

Conventional control theories with many ideal assumptions, such as synchronized control and nondelayed sensing and actuation, must be reevaluated before they can be applied to NCSs. For most applications, delays lead to a reduce of performances or can even lead to instability. However there exists some cases where the introduction of a delay in the control loop can help to stabilize a system which would not be stable without it (see [9], [10]).

In the present article, we will prove that the simple integrator consensus algorithm belongs to this class of systems and provide a novel consensus algorithm based on sampling approach. To do so, the classical consensus algorithm $\dot{x}(t) = -Lx(t)$ is modified into a new algorithm (see Figure 1), where $\delta \in \mathbb{R}$ and $\tau \geq 0$ are additional parameters. Note that if δ and/or τ are taken as zeros, then the classical algorithm is retrieved. As the delay is now a control parameter, we can choose it of the most appropriate form. In this article, we will consider a sampling delay, $\tau(t) = t - t_k,\ t_k \leq t < t_{k+1}$, already used in [11] or in [12] where the t_k's satisfies $0 = t_0 < t_1 < ... < t_k < ...$ corresponds to the sampling instants. For the sake of simplicity, we will assume that the sampling process is periodic, i.e. the difference between two successive sampling instants $t_{k+1} - t_k = T$, is constant.

From computational point of view, this choice is relevant with respect to the introduction of a constant delay τ since in the sampling delay case, only one data is held in the algorithm whereas in the case of a constant delay, all values of x in the interval $[t - \tau\, t]$ should be kept in memory.

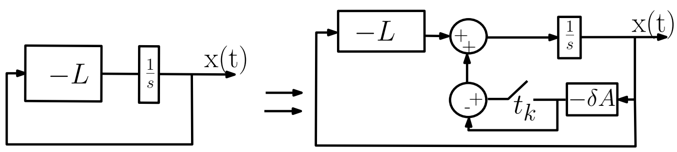


Fig. 1. Corresponding schema for the proposed algorithm.

A method to design the algorithm parameters, including the appropriated sampling period T, on a "optimal" way is proposed based on a LMI's formulation. The communication graphs are supposed to be directed and undirected.

This paper is organized as follows: Section 2 presents the problem treated in this article. Section 3 will be dedicated to the establishment of the appropriated model, and in Section 4 we will proceed to stability analysis of the algorithm. Section 5 includes illustrating simulation results and performance analysis, and finally, Section 6 will present our conclusions and indicate possible future research efforts.

This work was supported by the European Project FeedNetBack: http://www.feednetback.eu/.

Throughout the paper, the superscript 'T' stands for matrix transposition, $\mathbb{R}^n$ denote the n-dimensional Euclidean space, and $\mathbb{R}^{n\times m}$ is the set of $n \times m$ real matrices. The set $\mathbb{S}^n$ stands for the set of symmetric matrices of $\mathbb{R}^{n\times n}$. I represents the identity matrix. Finally, for any matrix M, the notation $(M)_i$ denotes the i^{th} line of M and $\lambda_k(M)$ represents the k^{th} eigenvalue of M. The notation $|.|$ corresponds the Euclidean norm and, for any function $\phi\ :\ [-\tau,\ 0]\ \to \mathbb{R}^n$, the notation $|\phi|_\tau$ corresponds to $\max_{\theta\in[-\tau,\ 0]} |\phi(\theta)|$.

For the graph G with N vertices and edge set given by $E = \{(i,j) : j \in \mathcal{N}_i\}$ the *adjacency matrix* $A = A(G) = (a_{ij})$ is the $N \times N$ matrix given by $a_{ij} = 1$, if $(i,j) \in E$ and $a_{ij} = 0$, otherwise. The *degree* d_i of vertex i is defined as the number of its neighboring vertices, i.e. $d_i = \#j : (i,j) \in E$. Let Δ be the $N \times N$ diagonal matrix of d_i's. The *Laplacian* of G is the matrix $L = \Delta - A$. For an undirected graph the Laplacian matrix is symmetric positive semidefinite. Zero is a simple eigenvalue of L (the corresponding eigenvector is the vector of ones, $\overrightarrow{\mathbf{1}}$) if and only if the associated directed graph has a directed spanning tree.

A. Preliminary lemmas and definitions

In order to clarify the presentation, a definition of exponential stability will be stated here.

Definition 1: ([13]) 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 $F \geq 1$ such that the solution $x(t;t_0,\phi)$ satisfies:

$$|x(t;t_0,\phi)| \leq F|\phi|_\tau e^{-\alpha(t-t_0)}. \tag{1}$$

In the sequel we will say that a system is α-stable if the solution of the system are exponentially stable with a decay rate α_g.

II. Problem Statement

In this paper the following problem in addressed. Consider the classical simple integrator consensus algorithm

$$\dot{x}_i(t) = \textstyle\sum_{j\in\mathcal{N}_i} a_{ij}(x_j(t_k) - x_i(t)) \quad i \in \{1,\dots,N\}, \tag{2}$$

where x represents the vector containing the agents variables. This algorithm is distributed in the sense that each vehicle needs only information from its neighbors. Moreover, consensus algorithms can be archived asymptotically if and only if the graph associated to the Laplacian L has a directed spanning tree (page 25 [14]). We then derive the control law:

$$\dot{x}(t) = -Lx(t)\ , \tag{3}$$

with $x(t) = [x_1(t),..,x_N(t)]^T$.

In this paper we will proposed an improved algorithm for simple integrator agents. The goal is a performance comparaison between the proposed and the classical algortihm, where memory's effects on system's stability will be bring forward. Assuming that there exists a constant and positive scalar μ such that:

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

the previous algorithm is modified into a new algorithm defined by

$$\dot{x}(t) = (-\mu I + (1-\delta)A)x(t) + \delta Ax(t_k), \tag{4}$$

where A is the adjacency matrix of the communication graph.

As said before, there exists some cases where the introduction of a delay in the control loop can have stabilizing properties.

Consider further that all the the sampling period is constant and equal to T. Then (4) can be written as:

$$\dot{x}(t) = (-L - \delta A)x(t) + \delta Ax(t_k) \tag{5}$$

By the Leibnitz formula, we have $x(t_k) = x(t) - \int_{t_k}^{t} \dot{x}(s)ds$, for all differentiable functions x. System (5) can be rewritten as:

$$\dot{x}(t) = -Lx(t) - \delta A \int_{t_k}^{t} \dot{x}(s)ds. \tag{6}$$

Note that the matrix $\mu I - A$ corresponds to the Laplacian matrix. This representation is a way to understand how the delay affects the consensus problem. From the point of view of agent i, the state x_i is available at every time t without any delay. However the data coming from the other agents $j \in \mathcal{N}_i$ are received by agent i after a time-delay (sampling period) chosen for control.

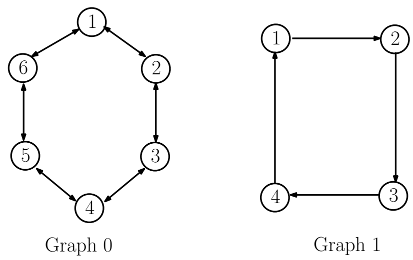


Fig. 2. Corresponding graphs of the matrices L_0 and L_1.

A. Definition of an appropriate model

This section focuses on the definition of a suitable modeling of the consensus algorithm (5) to analyze its convergence. Knowing that the vector $\overrightarrow{\mathbf{1}}$ is an eigenvector associated to the eigenvalue 0 of the Laplacian matrix, it is possible to find a change of coordinates $x = Wz$ such that:

$$U(-\mu I + A)W = \begin{bmatrix} B & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}, \tag{7}$$

where $U = \begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = W^{-1}$ and $U_2 = (U)_N$. In the case of a symmetric matrix A, the rest of the Laplacian eigenvalues are all positive. We denote them by $0 < \lambda_2 \leq \ldots \leq \lambda_N$. It thus means that $B \in \mathbb{R}^x$ is a diagonal matrix with $-\lambda_i$ for graphs containing a directed spanning tree.

The following lemma provides an appropriate way to rewrite (6) based on the properties of the matrix L.

Lemma 1: The system (6) can be rewritten in the following way:

$$\dot{z}_1(t) = (-B + \delta(B + \mu I))z_1(t) - \delta(B + \mu I)z_1(t_k), \quad \text{(8a)}$$
$$\dot{z}_2(t) = -\mu z_2(t) + \mu z_2(t_k), \quad \text{(8b)}$$

where $z_1 \in R^{N-1}$, $z_2 \in R$ and the matrix B in given in (11).

Proof: Consider system (6)

$$\begin{bmatrix} \dot{z}_1(t) \\ \dot{z}_2(t) \end{bmatrix} = -\begin{bmatrix} B & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} - \begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} \int_{t-\tau}^{t} \dot{z}(s)ds,$$

where $\begin{bmatrix} A'_1 \\ A'_2 \end{bmatrix} = UAW$ and $A'_2 = (UAW)_N$. We can then rewrite (5) into two equations where $z_1 = U_1 x \ \in \mathbb{R}^{(N-1)}$ and $z_2 = U_2 x \ \in \mathbb{R}^N$ represent respectively the $N-1$ first components and the last component of z. From (7), simple matrix calculations lead us to

$$\begin{bmatrix} A_{1t} \\ A_{2t} \end{bmatrix} = UAW = \begin{bmatrix} B + \mu I & \vec{0} \\ \vec{0}^T & \mu \end{bmatrix} \quad (9)$$

Using the Leibnitz formula, (5) can be rewritten as

$$\begin{array}{l} \dot{z}_1(t) = -Bz_1(t) + \delta(B + \mu I) \int_{t_k}^{t} \dot{z}_1(s)ds, \\ \dot{z}_2(t) = -\delta\mu \int_{t_k}^{t} \dot{z}_2(s)ds. \end{array} \quad (10)$$

■

The consensus problem is now expressed into an appropriate form to perform stability criteria. In the case of a symmetric network, the matrix W is an orthogonal matrix which means $U = W^T$. Then if the last column of W is $\beta\overrightarrow{\mathbf{1}}$, then $U_2 = 1/(\beta N)\overrightarrow{\mathbf{1}}$, which means that z_2 corresponds to the average of the position of all agents. This does not hold always for asymmetric communication network.

In the sequel, a stability analysis of the algorithm is proposed for any graph with a directed spanning tree, represented by the Laplacian L. Requiring a directed spanning tree is less stringent than requiring a strongly connected and balanced graph ([14]). Also, an inherent assumption is that all agents are synchronized and share the same clock. This analysis is composed by two parts, one dealing with the stability of the algorithm and another concerning the agreement of the agents. More particularly, we will propose a method to choose appropriately the algorithm parameters δ and T for a given L, considering a performance optimisation.

III. Does this algorithm always improve standard performances?

Assume that the Laplacian graph corresponds to a symmetric graph. Let B be the diagonal matrix of the Laplacian eigenvalues defined before. We know that

$$B = -\begin{bmatrix} \lambda_i & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_{N-1} \end{bmatrix}. \quad (11)$$

As seen previously, and for $i \in [0, N-1]$ we have

$$\dot{z}_{1i}(t) = (-\lambda_i + \delta(\lambda_i + \mu))z_{1i}(t) - \delta(\lambda_i + \mu I)z_{1i}(t_k). \quad (12)$$

For sake of simplicity, take $b = \delta(\lambda_i + \mu)$, and

$$\dot{z}_{1i}(t) = (-\lambda_i + b)z_{1i}(t) - bz_{1i}(t_k). \quad (13)$$

By integrating the previous equation, we can have the following recurrence equation

$$z_{1i}(t_{k+1}) = A(\lambda_i, \delta, T)z_{1i}(t_k), \quad (14)$$

with $A(\lambda_i, \delta, T) = \exp^{(-\lambda_i + b)T} \frac{-\lambda_i}{-\lambda_i + b} + \frac{b}{-\lambda_i + b}$. Note that system's (14) stability increases as $A(\lambda_i, \delta, T)$ decreases.

In this section we will prove that by varying δ and T values close to zero, we can achieve a performance improvement for $\forall \lambda_i$, if

$$\frac{\delta A(\lambda_i, \delta, T)}{\delta T} \leq 0, for\ some\ \delta\ values \quad (15a)$$
$$\frac{\delta A(\lambda_i, \delta, T)}{\delta \delta} \leq 0, for\ some\ T\ values \quad (15b)$$

From (14), by derivation of $A(\lambda_i, \delta, T)$, we have

$$\frac{\delta A(\lambda_i, \delta, T)}{\delta T} = \left[(-\lambda_i + b)e^{(-\lambda_i + b)T}\right] \frac{-\lambda_i}{(-\lambda_i + b)} \quad (16a)$$
$$\frac{\delta A(\lambda_i, \delta, T)}{\delta \delta} = T(\lambda_i + \mu)e^{(-\lambda_i + b)T} \left(\frac{-\lambda_i}{(-\lambda_i + b)}\right)$$
$$- e^{(\lambda_i + b)T} \left(\frac{-\lambda_i(\lambda_i + \mu)}{(-\lambda_i + b)^2}\right) + \frac{\lambda_i + \mu}{(-\lambda_i + b)} + \frac{b(\lambda_i + \mu)}{(-\lambda_i + b)^2} \quad (16b)$$

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

$$\frac{\delta A(\lambda_i, \delta, T)}{\delta T} = -\lambda_i \leq 0 \quad (17a)$$
$$\frac{\delta A(\lambda_i, \delta, T)}{\delta \delta} = e^{-\lambda_i T}(\lambda_i + \mu)(T + \frac{1}{\lambda_i}) - \left(\frac{\lambda_i + \mu}{\lambda_i}\right) \leq 0 \quad (17b)$$

As $\frac{\delta A(\lambda_i, \delta, T)}{\delta T} = -\lambda_i$ is negative semi-definite for all value of δ, and $\frac{\delta A(\lambda_i, \delta, T)}{\delta \delta}$ is also negative semi definite for small values of T, we can then conclude that for small values of δ and T system (14) tends to converge rapidly when compare with the trivial algorithm. The pertinent problem of how to chose theses parameters values has been rased here, and will be treated in the next section.

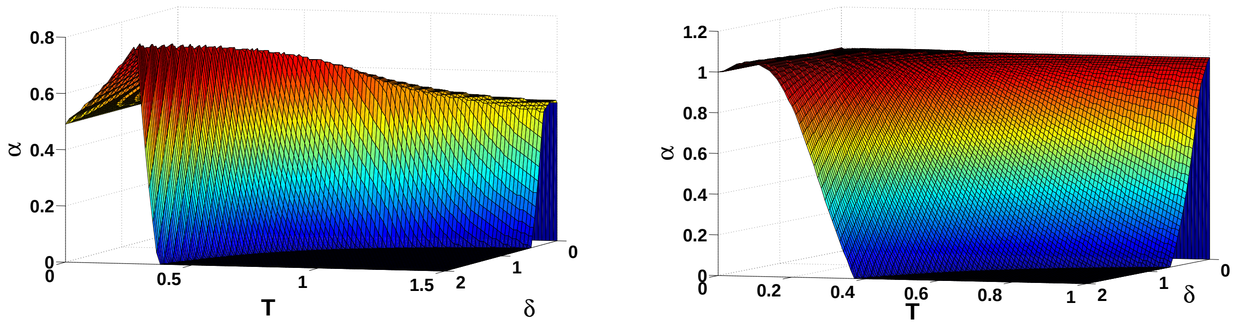
IV. Stability analysis

A. Preliminary stability analysis

This section deals with the stability analysis of (8b). The following lemma holds.

Lemma 2: The system defined in (8b) is stable for any sampling period T and any δ. The variable z_2 is constant

$$\forall t, \quad z_2(t) = z_2(0) \quad (18)$$



(a) Exponential decay rate for G_0

(b) Exponential decay rate for G_1

Fig. 3. Convergence rate of the consensus algorithm for several values of (δ, T) and for the communication graphs G_0 and G_1.

Proof: Consider $k \geq 0$ and any $t \in [t_k\ t_{k+1}[$ and any parameters T, δ. The previous ordinary differential equation has known solutions of the form

$$z_2(t) = e^{-\delta\mu(t-t_k)}C_0 - z_2(t_k) \tag{19}$$

where $C_0 \in \mathbb{R}$ represent the initial condition of the ordinary differential equation. The initial condition is determined at time $t = t_k$. We then obtain $C_0 = 0$ and thus

$$\forall t \in [t_k\ t_{k+1}[, \quad z_2(t) = z_2(t_k) \tag{20}$$

Then, we deduce

$$z_2(t_k) = z_2(0)$$

which means that z_2 is constant ■

B. Stability analysis of the consensus algorithm

Consider the consensus algorithm (5) rewritten in the form of (8). For z_1 dynamics analysis, we can establish

$$\dot{z}_1(t) = A(\delta)z_1(t) + A_d(\delta)\dot{z}_1(t_k), \tag{21}$$

with $A(\delta) = (-B + \delta(B + \mu I))$ and $A_d(\delta) = -\delta(B + \mu I)$. The following theorem holds based on the result of [12].

Theorem 1: Consider the proposed consensus algorithm (5) associated to a given Laplacian L representing a communication graph with a directed spanning tree, a given $\alpha > 0$, $\delta > 0$ and $T > 0$.

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

$$\Pi_1 + h_\alpha(T, 0)M_2^T X M_2 + f_\alpha(T, 0)\Pi_2 < 0, \tag{22}$$

$$\begin{bmatrix} \Pi_1 + h_\alpha(T,T)M_2^T X M_2 & g_\alpha(T,T)N \\ * & -g_\alpha(T,T)R \end{bmatrix} < 0, \tag{23}$$

where

$$\begin{array}{l} \Pi_1 = 2M_1^T P(M_0 + \alpha M_1) - M_3^T(S_1 M_3 + 2S_2 M_2) - 2NM_3 \\ \Pi_2 = M_0^T(RM_0 + 2S_1 M_3 + 2S_2 M_2), \end{array}$$

and $M_0 = \begin{bmatrix} A(\delta) & A_d(\delta) \end{bmatrix}$, $M_1 = \begin{bmatrix} I & 0 \end{bmatrix}$, $M_2 = \begin{bmatrix} 0 & I \end{bmatrix}$, $M_3 = \begin{bmatrix} I & -I \end{bmatrix}$. The functions f_α, g_α and h_α for all scalars T and $\tau \in [0\ T]$ are given by

$$\begin{array}{l} f_\alpha(T,\tau) = (e^{2\alpha(T-\tau)} - 1)/2\alpha, \\ g_\alpha(T,\tau) = e^{2\alpha T}(1 - e^{-2\alpha\tau})/2\alpha, \\ h_\alpha(T,\tau) = \frac{1}{\alpha}\left[\frac{e^{2\alpha T}-1}{2\alpha T} - e^{2\alpha\tau}\right]. \end{array} \tag{24}$$

Then, the consensus algorithm (5) with the parameter δ and the sampling period T is thus $\alpha-$stable. Moreover the consensus equilibrium is given by

$$x(\infty) = U_2 x(0). \tag{25}$$

Proof: Consider the consensus algorithm (5). Using Lemma 1, the algorithm is rewritten as (8). The stability of the second equation (8b) is ensured based on the discrete time Lyapunov theorem. Consider $\alpha \in \mathbb{R}$ and a positive definite matrix $P \in \mathbb{R}^{2N}$ which defines a Lyapunov function V for the discrete-time system defined by $x_k = x(t_k)$ given by $V(k) = x_k^T P x_k$. The objective is to ensure that the increment ΔV_α is negative definite ([15]):

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

We introduce here a novel type of functional for all $t \in [t_k\ t_{k+1}]$.

$$\begin{array}{rl} W_\alpha(t, x_t) = & f_\alpha(T,\tau)\zeta_0^T(t)[S_1\zeta_0(t) + 2S_2 x_k] \\ & +f_\alpha(T,\tau)\int_{t_k}^{t} \xi^T(s)M_0^T R M_0 \xi(s) ds \\ & +H_\alpha(T,\tau)x_k^T X x_k \end{array} \tag{26}$$

where $\zeta_0(t) = x(t) - x_k$, $\xi(s) = [x^T(s)\ \ x_k^T]^T$ and $H_\alpha(T,\tau) = \frac{e^{-2\alpha\tau}}{\alpha}\left[\frac{\tau}{T}\frac{e^{2\alpha T}-1}{2\alpha} - \frac{e^{2\alpha\tau}-1}{2\alpha}\right]$.
Denote $\bar{V}(t, x_t) = x^T(t)Px(t) + W_\alpha(t, x_t)$.

Consider a positive scalar $0 < \epsilon < T$ and the functional W_α at time $t_k - \epsilon$ and $t_k + \epsilon$. Since $\zeta_0(t_k + \epsilon)$ and $f_\alpha(T_{k-1}, T_{k-1} - \epsilon)$ tend to 0 as $\epsilon \to 0$ for all $\alpha > 0$, the following equalities are satisfied

$$\lim_{\epsilon\to 0} W_\alpha(t_k - \epsilon, x_{t_k-\epsilon}) = \lim_{\epsilon\to 0} W_\alpha(t_k + \epsilon, x_{t_k+\epsilon}) = 0. \tag{27}$$

This ensures that the extended functional $\bar{V}$ is continuous with respect to t at all sampling instants and differentiable over $[t_k\ t_{k+1}[$ and $W_\alpha(t_{k+1}, x_{t_{k+1}}) - W_\alpha(t_k, x_{t_k}) = 0$. No additional constraint is introduced on S_1 and S_2, W_α is not necessary positive definite within two sampling instants. It yields

$$\begin{aligned}\Delta_\alpha V(k) &= \bar{V}(t_{k+1}, x_{t_{k+1}}) - e^{-2\alpha T}\bar{V}(t_k, x_{t_k})\\ &= e^{-2\alpha T}\int_{t_k}^{t_{k+1}} d\left[e^{2\alpha\tau(s)}\bar{V}(s,x_s)\right]\\ &= e^{-2\alpha T}\int_{t_k}^{t_{k+1}} e^{2\alpha\tau(s)}(\dot{\bar{V}}(s,x_s)+2\alpha\bar{V}(s,x_s))ds\end{aligned}$$

The rest of the proof consists in ensuring $\tilde{V} = \dot{\bar{V}}(s,x_s) + 2\alpha\bar{V}(s,x_s) < 0$. From (24), we have, for all $\alpha \in \mathbb{R}$ and for all $\tau \in [0, T_k]$, $\dot{f}_\alpha(T,\tau) + 2\alpha f_\alpha(T,\tau) = -1$ and $\frac{d}{d\tau}(e^{2\alpha\tau}H_\alpha 4) = h_\alpha 4$. This leads to

$$\begin{aligned}\tilde{V}(t,x_t) &= 2x^T(t)P\dot{x}(t) - \zeta_0^T(t)[S_1\zeta_0(t) + 2S_2x_k]\\ &+2f_\alpha(T,\tau)\dot{x}(t)^T[RM_0\xi(t) + 2S_1\zeta_0(t) + 2S_2x_k]\\ &+h_\alpha(T,\tau)x_k^TXx_k + 2\alpha x^T(t)Px(t)\\ &-\int_{t_k}^t \xi^T(s)M_0^TRM_0\xi(s)ds.\end{aligned} \tag{28}$$

Consider a matrix $N \in \mathbb{R}^{2n\times n}$ and the following equality

$$2N\left[x(t) - x_k\right] = \int_{t_k}^t \left[2N\dot{x}(s)\right]ds = \int_{t_k}^t \left[2NM_0\xi(s)\right]ds. \tag{29}$$

Since $R > 0$ and consequently non singular, a classical bounding ensures that for all $t \in [t_k,\ t_{k+1}[$ and for all $s \in [t_k,\ t]$

$$2\xi^T(t)NM_0\xi(s) \le \xi^T(t)NR^{-1}N^T\xi(t) + \xi^T(s)M_0^TRM_0\xi(s).$$

Integrating the previous inequality over $[t_k,\ t]$, the following inequality is obtained

$$\begin{aligned}-\int_{t_k}^t \xi^T(s)M_0^TRM_0\xi(s)ds &\le -2\xi^T(t)N[x(t)-x_k]\\ &+\tau\xi^T(t)NR^{-1}N^T\xi(t),\end{aligned} \tag{30}$$

Noting that

$$\begin{aligned}&\dot{x}(t) = Ax(t) + A_dx(t_k) = M_0\xi(t), \quad x(t) = M_1\xi(t),\\ &x(t_k) = M_2\xi(t), \quad \zeta_0(t) = x(t) - x(t_k) = M_3\xi(t),\end{aligned}$$

and adding (30) to (28), the following inequality is obtained for all $t \in [t_k,\ t_{k+1}[$

$$\tilde{V}(t,x_t) \le \xi^T(t)[\Pi_1 + f_\alpha(T,\tau)\Pi_2 + \tau NR^{-1}N^T]\xi(t). \tag{31}$$

The previous inequality does not depend linearly on τ but on both τ and a non linear function of τ, $f_\alpha(T,\tau)$.

The solution proposed here is to use the convexity property of the exponential function ensuring that $e^{2\alpha\tau} \ge 1 + 2\alpha\tau$. Consequently, the following upper-bound is obtained

$$\tau \le (e^{2\alpha\tau} - 1)/2\alpha \le e^{2\alpha\tau}(1 - e^{-2\alpha\tau})/2\alpha, \le g_\alpha(T,\tau).$$

Since R, consequently R^{-1} is positive definite, we have

$$\tilde{V}(t,x_t) \le \xi^T(t)[\Pi_1 + f_\alpha(T,\tau)\Pi_2 + g_\alpha(T,\tau)NR^{-1}N^T]\xi(t).$$

To prove that $\tilde{V}$ is negative definite for all τ, a convexity property is employed. Since the previous inequality is linear with respect to τ, it is necessary and sufficient to ensure the negativity at the edges. This leads to

$$\begin{aligned}\Pi_1 + f_\alpha(T,0)\Pi_2 < 0,\\ \Pi_1 + g_\alpha(T,T)NR^{-1}N^T < 0.\end{aligned}$$

This leads to (22) and (23) using the Schur complement. The global algorithm (8) is then exponentially stable with a decay rate α. Moreover, $z_1 \to 0$ and $z_2 \to z_2(\infty)$, and thus the consensus equilibrium is

$$x(\infty) = Wz_2(\infty) = [W_1\ \overrightarrow{1}]\begin{bmatrix} 0_{n-1\times 1} \\ z_2(\infty)\end{bmatrix} = z_2(\infty)\overrightarrow{1}.$$

The proof is concluded by noting that $z_2 = U_2x$ and that $z_2(\infty)$ is expressed in (18) ■

V. EXAMPLES

Cooperative control of under-water vehicles network under varying-topology and communications constraints have been studied in order to implement the proposed algorithms. Consider a set of six and four agents connected through, respectively, the undirected and directed graphs shown in Figure 2.

To each graph is associated a Laplacian matrix given by

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

$$L_1 = \begin{bmatrix} -1 & 1 & 0 & 0\\ 0 & -1 & 1 & 0\\ 0 & 0 & -1 & 1\\ 1 & 0 & 0 & -1\end{bmatrix}.$$

For simulations purposes we took as initial conditions: $x_0^T(0) = [30\ 25\ 15\ 0\ -10\ -20]$ and $x_1^T(0) = [30\ 25\ 15\ 0]$. Those two graphs are balanced, witch implies that consensus equilibrium value will be defined as the average of initial conditions presented just before.

The objective is to find the highest value for α (on the vertical axis) that guarantees algorithm (5) convergence. Figure 3, as a 3-D representation of α stability results, shows the maximum convergence rate satisfying Theorem 1 for several values of δ and T, and for L_0 and L_1, with $T \in [0, 1.5]s$ and $\delta \in [0, 2]$. We can identified a crest for specific values of (δ, T) meaning a improved behavior, and the best positive value of α is obtained when (δ,T)=(2;0.32) and (δ,T)=(2;0.2), for graph G_0 and graph G_1 respectively.

The stability conditions proposed in this article are sufficient but not necessary conditions. Optimality is obtained for a certain value of (δ, T), and once it changes, this leads to a reduction of performances, as it will be shown in the following.

Figure (4) shows simulations of the classical algorithm (3) as well as from the algorithm (5) considering L_0, L_1, and for several values of δ and T. The aim here is to compare systems performances with two different approaches and justify the interest of the proposed algorithm. Figure 4(a-b) show simulation results of the classical consensus algorithm. Figure 4(c-d) show simulation results using the optimal pair (δ,T) according to Theorem 1 and recovered on Figure 3.We

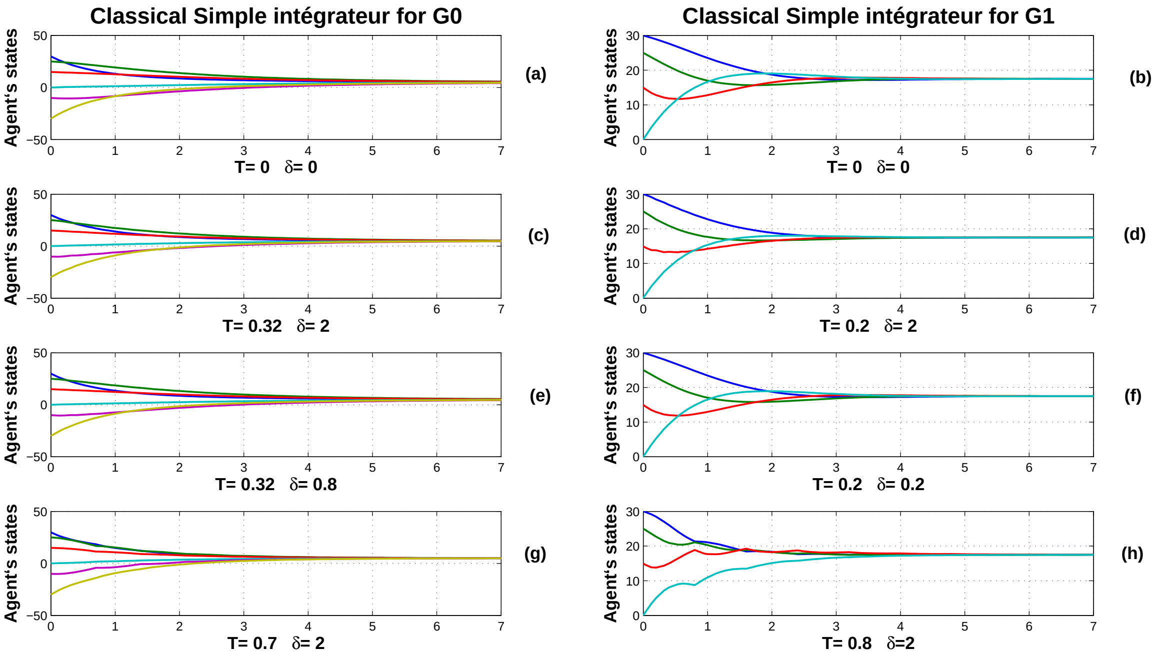


Fig. 4. System's behavior for different settings

can see that they correspond to a faster algorithm when compared with the trivial algorithm. In Figure 4(e-f), we kept the optimal value of T and changed δ value. Finally, for Figure 4(g-h), we kept the optimal value of δ and changed T value. In 4(c-d-e-f) we can then see that convergence rate decreases when compared to the others results. It's also possible to observe that the agreement value for the modified algorithm remains the average of the initial conditions.

VI. Conclusion

The influence of a sampled delays in consensus algorithms for simple integrator agents have been studied. An optimisation of controller parameters is proposed so that exponential stability of the solutions is achieved based on discrete-time Lyaponov Theorem and expressed in terms of LMI. Also, conditions for improved performances based on Laplacian's eigenvalues are provided here. Simulation results show the efficiency of the proposed algorithm, as well as the conservation of averaging properties. Further work might include asynchronous study of such systems.

References

[1] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *In EEE Trans.Automat. Contr.*, vol. 49, no. 8, pp. 1292–1302, 2004.

[2] R. Olfati-Saber and R. Murray, "Consensus problems in network of agents with switching topology and time delays," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, 2004.

[3] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42th IEEE Conference on Decision and Control*, 2003.

[4] D. Dimarogonas and K. Kyriakopoulos, "A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems," *Automatica*, vol. 44, no. 10, pp. 2648–2654, 2008.

[5] R. Olfati-Saber, A. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," in *Proceedings of IEEE*, vol. 95, 2007, p. 1.

[7] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," *In American Control Conference*, 2005.

[8] U. Muenz, A. Papachristodoulou, and F. Allgower, "Delay robustness in consensus problems," *To appear in Automatica*, 2010.

[9] K. Gu, V.-L. Kharitonov, and J. Chen, *Stability of time-delay systems*. Birkhauser, 2003.

[10] A. Seuret, C. Edwards, S. Spurgeon, and E. Fridman, "Static output feedback sliding mode control design via an artificial stabilizing delay," *IEEE Trans. on Automatic Control*, vol. 54, no. 2, pp. 256–265, 2007.

[11] G. Rodrigues de Campos and A. Seuret, "Continous-time double integrator consensus algortihms improved by an appropriate sampling," 2^{nd} *IFAC workshop on Distributed estimation and Control in Networked Systems*, 2010.

[12] A. Seuret, "A novel stability analysis of sampled-data systems with applications to multi-rate sampling and packet loss," *submitted to Automatica*, 2010.

[13] S.-I. Niculescu, C. E. D. Souza, L. Dugard, and J.-M. Dion, "Robust exponential stability and stabilization of uncertain systems with time-varying delays," *IEEE Trans. on Automatic control*, vol. 43, no. 5, pp. 743–748, 1998.

[14] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle cooperative Control: theory and applications*. Springer, 2008.

[15] E.I. Verriest and W. Michiels, "Stability analysis of linear systems with stochastically varying delays," *Systems and Control Letter*, vol. 58, no. 10-11, pp. 783–791, 2009.