



**HAL**  
open science

# Continuous-time double integrator consensus algorithms improved by an appropriate sampling

Gabriel Rodrigues de Campos, Alexandre Seuret

► **To cite this version:**

Gabriel Rodrigues de Campos, Alexandre Seuret. Continuous-time double integrator consensus algorithms improved by an appropriate sampling. NecSys 2010 - 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems, Sep 2010, Annecy, France. pp.017, 10.3182/20100913-2-FR-4014.00017 . hal-00515427

**HAL Id: hal-00515427**

**<https://hal.science/hal-00515427>**

Submitted on 6 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Continuous-time double integrator consensus algorithms improved by an appropriate sampling

Gabriel Rodrigues de Campos\* Alexandre Seuret,\*

\*NeCS Team, Automatic Control Department of Grenoble GIPSA-Lab,  
UMR CNRS 5216

E-mail:

gabriel.rodrigues-de-campos,alexandre.seuret@gipsa-lab.grenoble-inp.fr

---

**Abstract:** This paper deals with the double integrator consensus problem. The objective is the design of a new consensus algorithm for continuous-time multi-agent systems. The dynamic of agents is assumed to be of double integrator type. The proposed algorithm considers that there are no sensors to measure the velocity of the agents. Thus the classical double integrator consensus algorithm leads to an oscillatory behavior if the communication graph is undirected and to instability if the graph is directed. The novel algorithm proposes to sampled, in an appropriate manner, part of the multi-agent systems state such that the algorithm converges. An expression of the consensus equilibrium is provided. Some examples are provided to show the efficiency of the new algorithm.

Keywords: Consensus, Double Integrator Agents, Sampled Systems

---

### 1. INTRODUCTION

Network control systems (NCS) are spatially distributed systems with a communication network used between sensors, actuators, and controllers, which allows flexible architectures and generally reduces installation and maintenance costs, and identifies control over networks as one of the key future directions for control. This means NCS's applications can be found in a large range of areas such as mobile sensor networks (Ogren et al. [2004]), remote surgery, haptic collaboration over Internet, multi-robot systems (Olfati-Saber and Murray [2004]), automated highway systems, averaging in communication networks (Xiao and Boyd [2003]) and formation control (Dimarogonas and Kyriakopoulos [2008]). 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 Olfati-Saber et al. [2007], Hespanha et al. [2007] and Ren et al. [2005].

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, which can dramatically affect "consensus" convergence rate and cooperative control laws efficiency (Muenz et al. [2010]). Here, we consider that agents are assumed to obey a double integrator model. This model fits the behavior of real robotic agents more naturally, since such mechanical systems are controlled in most cases through their acceleration and not their velocity. Moreover, several robotic systems can be reduced to a double

integrator via a transformation in their control law. But double integrator algorithms leads to several problems (Ren [2008]), (Ren and Cao [2008]). For instance, if the graph is directed, the algorithm is not stable, and on the other hand, if the graph is undirected, this requires the knowledge of both position and velocity to converge to the same value. In this paper, we provide a novel consensus algorithm based on sampling approach, since it's known that for some systems delays have a stabilizing property (Michiels et al. [2004]), and sampling can be seen as a particular delay,  $\tau(t) = t - t_k$  (Fridman et al. [2004]). 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. The new algorithm offers several advantages, since it drastically reduce information quantity needed for control or quantity of sensors needed, which means economical, space and calculation savings.

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.

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  $i^{\text{th}}$  line of  $M$  and  $\lambda_k(M)$  represents the  $k^{\text{th}}$  eigenvalue of  $M$ . The notation  $|\cdot|$  corresponds the Euclidean norm and, for any function  $\phi : [-\tau, 0] \rightarrow \mathbb{R}^n$ , the notation  $|\phi|_\tau$  corresponds to  $\max_{\theta \in [-\tau, 0]} |\phi(\theta)|$ .

---

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

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. If there is an edge connecting two vertices  $i, j$ , i.e.  $(i, j) \in E$ , then  $i, j$  are called *adjacent*. If there is a path between any two vertices of the graph  $G$ , then  $G$  is called *strongly connected* in the case of directed, and simply *connected* in the case of undirected graphs. 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  $\Delta$  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,  $\vec{1}$ ) if and only if the associated directed graph has a directed spanning tree. This implies that the algorithm reaches consensus if and only if the directed communication topology has a directed spanning tree or the undirected communication topology is connected.

### 1.1 Preliminary lemmas and definitions

In order to clarify the presentation, a lemma on positivity of matrix inequations and a definition of exponential stability will be stated here.

*Lemma 1.* (Naghshtabrizi et al. [2008]) Consider three matrices  $X_1, X_2$  and  $X_3 \in \mathbb{S}^n$  and a time-varying parameter  $\lambda : \mathbb{R}^+ \rightarrow [\lambda_m, \lambda_M]$ , for some given  $\lambda_m$  and  $\lambda_M$ . If the following inequality is guaranteed

$$\forall t \geq 0, \quad X_1 + (\lambda_M - \lambda(t))X_2 + (\lambda(t) - \lambda_m)X_3 < 0, \quad (1)$$

then, it is equivalent to

$$X_1 + (\lambda_M - \lambda_m)X_2 < 0, \quad X_1 + (\lambda_M - \lambda_m)X_3 < 0. \quad (2)$$

*Definition 1.* (Niculescu et al. [1998]) Let  $\alpha > 0$  be some positive, constant, real number. The system is said to be exponentially stable with the decay rate  $\alpha$ , or  $\alpha$ -stable, if there exists a scalar  $F \geq 1$  such that the solution  $x(t; t_0, \phi)$  satisfies:

$$|x(t; t_0, \phi)| \leq F|\phi|_\tau e^{-\alpha(t-t_0)}. \quad (3)$$

In the sequel we will say that a system is  $\alpha$ -stable if the solution of the system are exponentially stable with a decay rate  $\alpha_g$ .

## 2. PROBLEM STATEMENT

Consider the classical double integrator consensus algorithm

$$\ddot{x}(t) = -\sigma\dot{x}(t) - Lx(t), \quad (4)$$

where  $x$  represents the vector containing the agents variables. From the literature (Ren [2008]), it can be seen that such algorithms with  $\sigma > 0$  lead to an agreement of the agents if the communication graph is undirected but the algorithm is not necessarily stable if the graph associated to the Laplacian  $L$  is directed. The case  $\sigma = 0$  leads to additional difficulties since we have:

$$\ddot{x}(t) = -Lx(t), \quad (5)$$

or, by introducing the augmented vector  $y(t) = [x^T(t) \dot{x}^T(t)]^T$ ,

$$\dot{y}(t) = \begin{bmatrix} 0 & I \\ -L & 0 \end{bmatrix} y(t) = \bar{L}y(t), \quad (6)$$

Noting that the trace of the matrix is zero whatever the communication graph, the eigenvalues of  $L$  are either on the imaginary axis or there is at least one eigenvalue on the right side of the imaginary axis. This leads to an oscillatory or unstable behavior of the algorithm.

However it is important to put forward the technical advantages of this assumption: by supposing  $\sigma = 0$ , we reduce drastically the information quantity needed for the control laws, and in a technical point of view, no velocity sensors are needed but only sensors to get the agent's position. This means economical, space and calculation savings.

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 Gu et al. [2003], Seuret et al. [2007]). In the present article, we will prove that the double integrator consensus algorithm belongs to this class of systems. To do so, the previous algorithm is modified into a new algorithm defined by

$$\ddot{x}(t) = -(L + \delta^2 I)x(t) + \delta^2 x(t - \tau) \quad (7)$$

where  $\delta \in \mathbb{R}$  and  $\tau \geq 0$  are additional parameters. Note that if  $\delta$  and/or  $\tau$  are taken as zeros, then the classical algorithm is retrieved. If  $\delta$  and  $\tau$  are not zero, then one can see that the proposed algorithm can be explained as follows. The diagonal contribution of the Laplacian is split into two parts: one delayed and the other is kept at the current time. This allows conserving the averaging properties of the agreement algorithm.

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 that was used in Fridman et al. [2004] or in Seuret [2010]

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

where the  $t_k$ 's satisfies  $0 = t_0 < t_1 < \dots < t_k < \dots$  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  $\tau$  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. Finally the proposed algorithm is

$$\forall t \in [t_k, t_{k+1}[, \quad \ddot{x}(t) = -(L + \delta^2 I)x(t) + \delta^2 x(t_k). \quad (8)$$

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 (W. Ren and R. W. Beard [2008]). 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  $\delta$  and  $T$  for a given  $L$ , considering a performance optimisation.

## 3. DEFINITION OF AN APPROPRIATE MODEL

This section focuses on the definition of a suitable modeling of the consensus algorithm (8) to analyze its convergence.

Knowing that the vector  $\vec{1}$  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

$$ULW = \begin{bmatrix} \Delta & 0 \\ 0^T & 0 \end{bmatrix}, \quad (9)$$

where  $\Delta \in \mathbb{R}^x$ , and for graphs containing a directed spanning tree,  $U = [U_1^T U_2^T]^T = W^{-1}$  and  $U_2 = (U)_N$  corresponds to the  $N^{th}$  line of  $U$ . If  $L$  represents a disconnected graph, the dimension of the vectors  $z_1$  and  $z_2$  change. The next lemma shows an appropriate way to rewrite (8) based on the properties of  $L$ .

**Lemma 2.** The consensus problem (8) can be rewritten using  $z_1 \in \mathbb{R}^{N-1}$ ,  $z_2 \in \mathbb{R}$  and the matrix  $\Delta$  is given in (9):

$$\dot{z}_1(t) = -(\Delta + \delta^2 I)z_1(t) + \delta^2 z_1(t_k), \quad (10a)$$

$$\dot{z}_2(t) = -\delta^2 z_2(t) + \delta^2 z_2(t_k), \quad (10b)$$

**Proof.** Consider (8) and note that it can be rewritten as follows

$$\forall t \in [t_k, t_{k+1}[, \quad \dot{x}(t) = -Lx(t) - \delta^2 \int_{t_k}^t \dot{x}(s)ds$$

Applying the change of coordinates  $z = Ux$ , and from (9), (8) is rewritten 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  $z$ . Noting that  $U(-\delta^2 I)W = -\delta^2 I$ , (8) can be rewritten using the Leibnitz formula as

$$\dot{z}_1(t) = -\Delta z_1(t) - \delta^2 \int_{t_k}^t \dot{z}_1(s)ds, \quad (11)$$

$$\dot{z}_2(t) = -\delta^2 \int_{t_k}^t \dot{z}_2(s)ds.$$

The proof is concluded using  $\int_{t_k}^t \dot{z}_i(s)ds = z_i(t) - z_i(t_k)$ . ■

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  $\beta \vec{1}$ , then  $U_2 = 1/(\beta N) \vec{1}$ , which means that  $z_2$  corresponds to the average of the position of all agents. This does not hold always for asymmetric communication network.

Regarding the stability of  $z_1$ , we introduce the augmented vector  $y = [z_1^T(t) \dot{z}_1^T(t)]^T$ . Then the dynamics of  $z_1$  can be rewritten as follows

$$\dot{y}(t) = A(\delta)y(t) + A_d(\delta)y(t_k),$$

$$\text{where } A(\delta) = \begin{bmatrix} 0 & I \\ -(\Delta + \delta^2 I) & 0 \end{bmatrix} \text{ and } A_d(\delta) = \begin{bmatrix} 0 & 0 \\ \delta^2 I & 0 \end{bmatrix}.$$

## 4. STABILITY ANALYSIS

### 4.1 Preliminary stability analysis

This section deals with the stability analysis of (10b). The following lemma holds.

**Lemma 3.** The system defined in (10b) is stable for any sampling period  $T$  and any  $\delta$  such that  $\delta T \neq 0 [\pi]$ , i.e  $\delta T \neq k\pi$ . The variable  $z_2$  converges to

$$z_2(\infty) = z_2(0) + \gamma_{\delta T} \dot{z}_2(0) \quad (12)$$

where  $\gamma_{\delta T} = \sin(\delta T)/(\delta(1 - \cos(\delta T))) = \tan((\pi - \delta T)/2)/\delta$ . Moreover the convergence rate of the solution to this equilibrium is  $-\log|\cos(\delta T)|$ .

**Proof.** Consider  $k \geq 0$  and any  $t \in [t_k, t_{k+1}[$  and any parameters  $T, \delta$  such that  $\delta T \neq 0 [\pi]$ . Define the augmented vector  $\bar{y}(t) = [z_2^T(t) \dot{z}_2(t)]^T$ , (10b) can be rewritten as follows

$$\dot{\bar{y}}(t) = B\bar{y}(t) + B_d\bar{y}(t_k) \quad (13)$$

where  $B = \begin{bmatrix} 0 & 1 \\ -\delta^2 & 0 \end{bmatrix}$  and  $B_d = \begin{bmatrix} 0 & 0 \\ \delta^2 & 0 \end{bmatrix}$ . It is easy to see that  $B$  is invertible and that  $B^{-1} = \begin{bmatrix} 0 & -1/\delta^2 \\ 1 & 0 \end{bmatrix}$ . The previous ordinary differential equation has known solutions of the form

$$\bar{y}(t) = e^{B(t-t_k)}[C_0 \ C_1]^T - B^{-1}B_d\bar{y}(t_k) \quad (14)$$

where  $C_0$  and  $C_1 \in \mathbb{R}$  represent the initial conditions of the ordinary differential equation. This leads to

$$\bar{y}(t) = \begin{bmatrix} \cos(w(t)) & \sin(w(t))/\delta \\ -\delta \sin(w(t)) & \cos(w(t)) \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \bar{y}(t_k) \quad (15)$$

where  $w(t) = \delta(t - t_k)$ . The initial conditions are determined at time  $t = t_k$ . We then obtain  $C_0 = 0$  and  $C_1 = [0 \ 1] \bar{y}(t_k)$ . Thus, we get the recurrence equation

$$\bar{y}(t_{k+1}) = \begin{bmatrix} 1 & \sin(\delta T)/\delta \\ 0 & \cos(\delta T) \end{bmatrix} \bar{y}(t_k) = \begin{bmatrix} 1 & \sin(\delta T)/\delta \\ 0 & \cos(\delta T) \end{bmatrix}^{k+1} \bar{y}(0)$$

where  $T = t_{k+1} - t_k$ . Simple computations lead to

$$\bar{y}(t_{k+1}) = \begin{bmatrix} 1 & \sin(\delta T) \sum_{i=0}^{k+1} \cos(\delta T)^i / \delta \\ 0 & \cos(\delta T)^{k+1} \end{bmatrix} \bar{y}(0) \quad (16)$$

In the previous expression, we recognize a geometric sequence. Thus, we obtain the following expression

$$\bar{y}(t_{k+1}) - \begin{bmatrix} z_2(\infty) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & -\gamma_{\delta T} \cos(\delta T) \\ 0 & 1 \end{bmatrix} \cos(\delta T)^{k+1} \bar{y}(0) \quad (17)$$

The assumption on  $\delta T \neq 0 [\pi]$  implies  $\cos(\delta T) < 1$ . Then it implies that  $z_2(t_k)$  tends zero and  $z_2$  to  $z_2(\infty)$  defined in (12). This concludes the proof. ■

### 4.2 Stability analysis of the consensus algorithm

Consider the consensus algorithm (8) rewritten in the form of (10). The following theorem holds based on the result of Seuret [2010]

**Theorem 1.** Consider the proposed consensus algorithm (8) associated to a given Laplacian  $L$  representing a communication graph with a directed spanning tree, a given  $\alpha > 0$ ,  $\delta > 0$  and  $T > 0$  such that  $\delta T \neq 0 [\pi]$ .

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

$$\Pi_1 + h_\alpha(T, 0)M_2^T X M_2 + f_\alpha(T, 0)\Pi_2 < 0, \quad (18)$$

$$\begin{bmatrix} \Pi_1 + h_\alpha(T, T)M_2^T X M_2 & g_\alpha(T, T)N \\ * & -g_\alpha(T, T)R \end{bmatrix} < 0, \quad (19)$$

where

$$\Pi_1 = 2M_1^T P(M_0 + \alpha M_1) - M_3^T (S_1 M_3 + 2S_2 M_2) - 2N M_3$$

$$\Pi_2 = M_0^T (R M_0 + 2S_1 M_3 + 2S_2 M_2),$$

and  $M_0 = [A(\delta) \ A_d(\delta)]$ ,  $M_1 = [I \ 0]$ ,  $M_2 = [0 \ I]$ ,  $M_3 = [I \ -I]$ . The functions  $f_\alpha$ ,  $g_\alpha$  and  $h_\alpha$  for all scalars  $T$  and  $\tau \in [0 \ T]$  are given by

$$\begin{aligned} f_\alpha(T, \tau) &= (e^{2\alpha(T-\tau)} - 1)/2\alpha, \\ g_\alpha(T, \tau) &= e^{2\alpha T} (1 - e^{-2\alpha\tau})/2\alpha, \\ h_\alpha(T, \tau) &= -1 - 2\alpha f_\alpha(T, \tau) - f_\alpha(T, 0)/T. \end{aligned} \quad (20)$$

Then, the consensus algorithm (8) with the parameter  $\delta$  and the sampling period  $T$  is thus  $\alpha_g$ -stable, where

$\alpha_g = \min\{\alpha, -\log(\cos(\delta T))\}$ . Moreover the consensus equilibrium is given by

$$x(\infty) = U_2(x(0) + \gamma_{\delta T} \dot{x}(0)). \quad (21)$$

**Proof.** Consider the consensus algorithm (8). Using Lemma 2, the algorithm is rewritten as (10). The stability of the second equation (10b) 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  $y_k = y(t_k)$  given by  $V(k) = y_k^T P y_k$ . The objective is to ensure that the increment  $\Delta V_\alpha$  is negative definite (E.I. Verriest and W. Michiels [2009]):

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

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

$$\begin{aligned} W_\alpha(t, y_t) &= f_\alpha(T, \tau) \zeta_0^T(t) [S_1 \zeta_0(t) + 2S_2 y_k] \\ &+ f_\alpha(T, \tau) \int_{t_k}^t \xi^T(s) M_0^T R M_0 \xi(s) ds \\ &+ (f_\alpha(T, 0) - f_\alpha(T, \tau) - \tau/T f_\alpha(T, 0)) y_k^T X y_k \end{aligned} \quad (22)$$

where  $\zeta_0(t) = y(t) - y_k$ ,  $\xi(s) = [y^T(s) y_k^T]^T$ .

Denote  $\bar{V}(t, y_t) = y^T(t) P y(t) + W_\alpha(t, y_t)$ . Consider a positive scalar  $0 < \varepsilon < T$  and the functional  $W_\alpha$  at time  $t_k - \varepsilon$  and  $t_k + \varepsilon$ . Since  $\zeta_0(t_k + \varepsilon)$  and  $f_\alpha(T, T_{k-1} - \varepsilon)$  tend to 0 as  $\varepsilon \rightarrow 0$  for all  $\alpha > 0$ , the following equalities are satisfied

$$\lim_{\varepsilon \rightarrow 0} W_\alpha(t_k - \varepsilon, y_{t_k - \varepsilon}) = \lim_{\varepsilon \rightarrow 0} W_\alpha(t_k + \varepsilon, y_{t_k + \varepsilon}) = 0. \quad (23)$$

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  $W_\alpha(t_{k+1}, y_{t_{k+1}}) - W_\alpha(t_k, y_{t_k}) = 0$ . No additional constraint is introduced on  $S_1$  and  $S_2$ ,  $W_\alpha$  is not necessary positive definite within two sampling instants. Then we obtain

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

The rest of the proof consists in ensuring  $\dot{\bar{V}} = \dot{\bar{V}}(s, y_s) + 2\alpha \bar{V}(s, y_s) < 0$ . From (20), we have, for all  $\alpha \in \mathbb{R}$  and for all  $\tau \in [0, T_k]$ ,  $\dot{f}_\alpha(T, \tau) + 2\alpha f_\alpha(T, \tau) = -1$ . This leads to

$$\begin{aligned} \dot{\bar{V}}(t, y_t) &= 2y^T(t) P \dot{y}(t) - \zeta_0^T(t) [S_1 \dot{\zeta}_0(t) + 2S_2 \dot{y}_k] \\ &+ 2f_\alpha(T, \tau) \dot{y}(t)^T [R M_0 \xi(t) + 2S_1 \zeta_0(t) + 2S_2 y_k] \\ &+ h_\alpha(T, \tau) y_k^T X \dot{y}_k + 2\alpha y^T(t) P y(t) \\ &- \int_{t_k}^t \xi^T(s) M_0^T R M_0 \xi(s) ds. \end{aligned} \quad (24)$$

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

$$2N[y(t) - y_k] = \int_{t_k}^t [2N\dot{y}(s)] ds = \int_{t_k}^t [2NM_0 \xi(s)] ds. \quad (25)$$

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

$$2\xi^T(t) N M_0 \xi(s) \leq \xi^T(t) N R^{-1} N^T \xi(t) + \xi^T(s) M_0^T R M_0 \xi(s).$$

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

$$\begin{aligned} - \int_{t_k}^t \xi^T(s) M_0^T R M_0 \xi(s) ds &\leq -2\xi^T(t) N [y(t) - y_k] \\ &+ \tau \xi^T(t) N R^{-1} N^T \xi(t), \end{aligned} \quad (26)$$

Noting that

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

and adding (26) to (24), the following inequality is obtained for all  $t \in [t_k, t_{k+1}[$

$$\dot{\bar{V}}(t, y_t) \leq \xi^T(t) [\Pi_1 + f_\alpha(T, \tau) \Pi_2 + \tau N R^{-1} N^T] \xi(t). \quad (27)$$

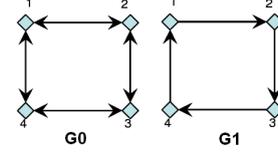


Fig. 1. Corresponding graphs of the matrices  $L_0$  and  $L_1$ .

The previous inequality does not depend linearly on  $\tau$  but on both  $\tau$  and a non linear function of  $\tau$ ,  $f_\alpha(T, \tau)$ . The solution proposed here is to use the convexity property of the exponential function ensuring that  $e^{2\alpha\tau} \geq 1 + 2\alpha\tau$ . Consequently, the following upper-bound is obtained

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

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

$$\bar{V}(t, y_t) \leq \xi^T(t) [\Pi_1 + f_\alpha(T, \tau) \Pi_2 + g_\alpha(T, \tau) N R^{-1} N^T] \xi(t).$$

To prove that  $\bar{V}$  is negative definite for all  $\tau$  we apply Lemma 1 with  $\lambda(t) = e^{-2\alpha\tau}$ , which leads to

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

This leads to (18) and (19) using the Schur complement. Thus, (10a) and (10b) are exponentially stable with a respectively decay rate  $\alpha$  and  $-\log|\cos(\delta T)|$ . The global algorithm (10) is then exponentially stable with the decay rate  $\alpha_g$ . Moreover, according to Lemma 2, and since  $z_1 \rightarrow 0$  and  $z_2 \rightarrow z_2(\infty)$ , we conclude that the consensus algorithm is exponentially stable with the consensus equilibrium being

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

The proof is concluded by noting that  $z_2 = U_2 x$  and that  $z_2(\infty)$  is expressed in (12) ■

The previous calculations state stability conditions for the continuous system (8) in terms of LMIs. However, Theorem (1) is based on the discrete-time Lyapunov theorem.

#### 4.3 Design of Control Parameters

In this article, performance analysis is based on a desired criteria, meaning the maximisation of  $\alpha_g$  in order to ensure system's exponential stability with the maximum decay rate. Then, for a reasonable set of chosen values for control parameter  $\delta$  and  $T$ , the optimal value of  $\alpha_g$  have been calculated for each point  $(\delta, T)$ . We would also like to discuss here the influence of initial conditions on the consensus equilibrium presented in (21). If the objective is to make agents converge to a common value, which depends on both initial positions and velocities, the initial velocities  $\dot{x}(0)$  may thus corrupt the final equilibrium in case of distributed estimation:

$$x(\infty) = U_2(x(0) + \Delta_z) \quad (28)$$

where  $\Delta_z = \gamma_{\delta T} \dot{x}(0)$ . Thus, there exist a way to ensure that the initial velocities will not corrupt dramatically the consensus equilibrium. With this approach, it is possible to choose  $\delta$  and  $T$  such that  $\Delta_z < \varepsilon$ , where  $\varepsilon$  is a given bound of the estimation error. More precisely, by choosing the good values for  $\delta$  and  $T$ , we can control the consensus equilibrium error  $\varepsilon$ .

## 5. EXAMPLES

In the framework of the European Project FeedNetBack, cooperative control of a four under-water vehicles network under

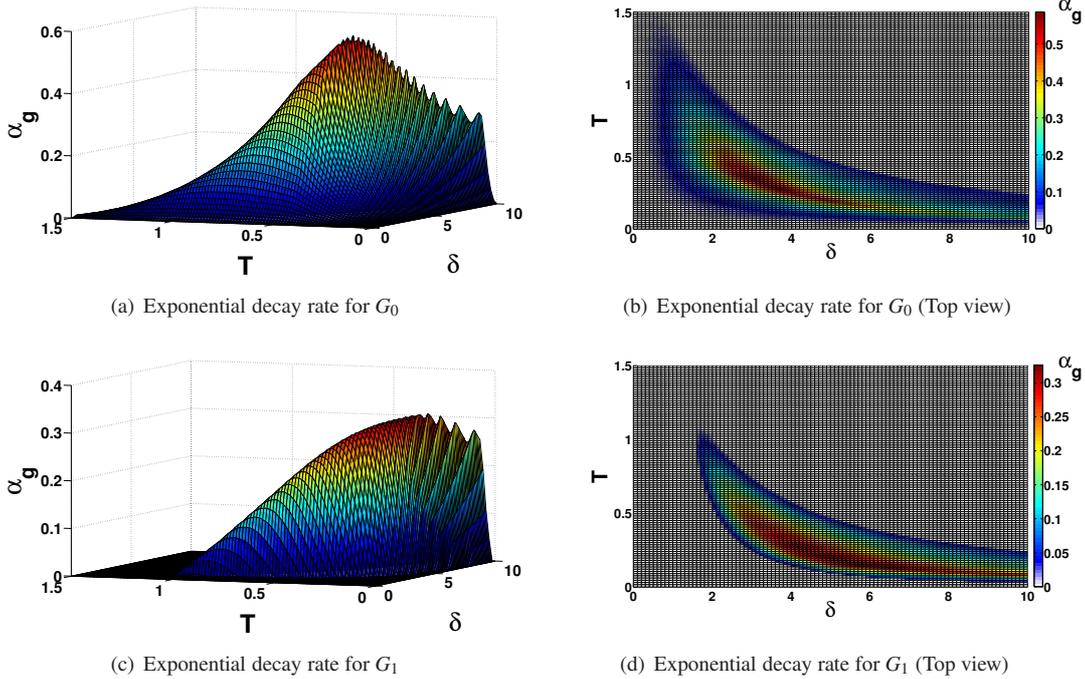


Fig. 2. Convergence rate of the consensus algorithm for several values of  $(\delta, T)$  and for the communication graphs  $G_0$  and  $G_1$ .

varying-topology and communications constraints is studied. Consider a set of four agents connected through the undirected and directed graphs shown in Figure 1.

To each graph is associated a Laplacian matrix given by

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

and for simulations purposes we took as initial conditions:  $x^T(0) = [20 \ 15 \ 5 \ 0]$  and  $\dot{x}^T(0) = [1 \ 2 \ 3 \ 2]$ . Those two graphs are balanced, which implies that consensus equilibrium value will be defined as the average of initial conditions presented just before weighted by different values of  $\delta$  and  $T$ .

The objective is to find the highest value for  $\alpha_g$  (on the vertical axis) that guarantees algorithm (8) convergence. Figure 2 shows the maximum convergence rate  $\alpha_g$  satisfying Theorem 1 for several values of  $\delta$  and  $T$ , and for  $L_0$  and  $L_1$ , with  $T \in [0, 1.5]s$  and  $\delta \in [0, 10]$ . In Figure 2(a),2(c) are a 3-D representation of  $\alpha_g$  stability results. Figure 2(b),2(d) show a top-view from the previous figures allowing us to observe  $\alpha_g$  values space distribution. On these figures, we observe a region where  $\alpha_g > 0$  which corresponds to the stability region of (8). The best positive value of  $\alpha_g$  is obtained when  $(\delta, T) = (3; 0.3)$  and  $(\delta, T) = (6.4; 0.15)$ , for graph  $G_0$  and graph  $G_1$  respectively. The fact that for a specific value of  $\delta$  and  $T$ , stability is not guaranteed by Theorem 1, does not necessary mean that the algorithm is unstable: in Figure 3(e), the algorithm is stable even though under our conditions we consider it unstable. The stability conditions proposed in this article are sufficient but not necessary conditions. Optimality is obtained for a certain value of  $T$ , and once it becomes too small or too significant, this leads to a reduction of performances, as it will be shown in the following.

Figure (3) shows simulations of algorithm (8) considering  $L_0$ ,  $L_1$ , and different values of  $\delta$  and  $T$ . If  $T = 0$ , this algorithm is unstable for directed and undirected graph 3(a-b), with an oscillating behavior for the undirected graph 3(a). Figure 3(c-d) show simulation results using the optimal pair  $(\delta, T)$  according to Theorem 1 (and Figure 2). We can see that they correspond to the fastest algorithm presented here. Figure 3(e) show a stable response but with a greater convergence rate when compared with Figure 3(c). In Figures 3(f-g) we can find the particular case where  $T = \pi/\delta$ , which does not fulfill assumptions of Theorem 1. Instability can be seen, with oscillations around the consensus algorithm final value. Finally, Figures 3(h-i-j) show algorithm behavior for greater values of  $T$ . It's easy to conclude that algorithm's performances decrease, and become unstable. In Table 1 we can find  $\gamma_{\delta T}$  and  $x(\infty)$  values for different  $\delta$  and  $T$ . For the particular case where  $T = \pi/\delta$  we can observe that consensus final value corresponds to the average of position's initial conditions ( $\gamma_{\delta T} = 0$ ), but no stability is achieved. The optimal behavior does not corresponds to the point where we find the smallest error  $\epsilon$ , but where decay rate  $\alpha_g$  is the greatest.

## 6. CONCLUSION

In this article, we proposed a new consensus algorithm for double integrator agents. This paper puts forward the technical advantages of such algorithm, since it reduces information quantity needed for control, as no more need of velocity sensors, meaning economical, space and calculation savings. An optimisation of controller parameters is proposed so that exponential stability of the solutions is achieved. An expression of the consensus equilibrium is derived so that the influence of the initial position with respect to the initial velocity can be achieved. A constraint on the proposed stability criteria expressed in term of LMIs is that the complexity will drastically increase for large networks. Thus, it is pertinent to develop other tools to analyse this algorithm for larger agents networks.

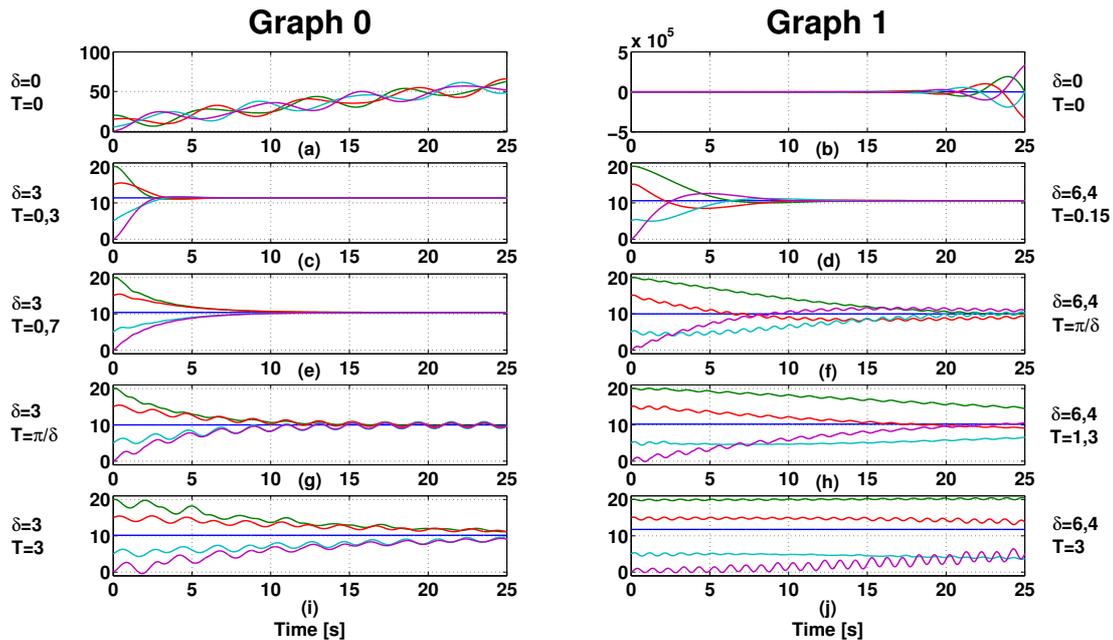


Fig. 3. Evolution of the agents state for several values of the sampling period  $T$

Graph 0					
$\delta = 3$					
T	0	0,3	0,7	$\pi/\delta$	3
$\gamma_{\delta T}$	222,22	0,69	0,19	0	0,07
$x(\infty)$	454,44	11,38	10,38	10	10,14
Graph 1					
$\delta = 6,4$					
T	0	0,15	$\pi/\delta$	1,3	3
$\gamma_{\delta T}$	48,82	0,30	0	0,09	0,88
$x(\infty)$	107,65	10,60	10	10,19	11,76

Table 1.

## REFERENCES

- D.V. Dimarogonas and K.J. Kyriakopoulos. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems. *Automatica*, 44(10):2648–2654, 2008.
- E.I. Verriest and W. Michiels. Stability analysis of linear systems with stochastically varying delays. *Systems and Control Letter*, 58(10-11):783–791, 2009.
- E. Fridman, A. Seuret, and J.-P. Richard. Robust sampled-data stabilization of linear systems: An input delay approach. *Automatica*, 40(8):1141–1146, 2004.
- K. Gu, V.-L. Kharitonov, and J. Chen. *Stability of time-delay systems*. Birkhauser, 2003.
- J. P. Hespanha, P. Naghshtabrizi, and Y. Xu. A survey of recent results in networked control systems. In *Proceedings of IEEE*, volume 95, page 1, 2007.
- W. Michiels, S.-I. Niculescu, and L. Moreau. Using delays and time-varying gains to improve the static output feedback stabilizability of linear systems : a comparison. *IMA Journal of Mathematical Control and Information*, 21(4):393–418, 2004.
- U. Muenz, A. Papachristodoulou, and F. Allgower. Delay robustness in consensus problems. *To appear in Automatica*, 2010.
- P. Naghshtabrizi, J.P. Hespanha, and A.R. Teel. Exponential stability of impulsive systems with application to uncertain sampled-data systems. In *Systems and Control Letters*, 57(5):378–385, 2008.
- S.-I. Niculescu, C. E. De Souza, L. Dugard, and J.-M. Dion. Robust exponential stability and stabilization of uncertain systems with time-varying delays. *IEEE Trans. on Automatic control*, 43(5):743–748, 1998.
- P. Ogren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. In *IEEE Trans. Automat. Contr.*, 49(8):1292–1302, 2004.
- R. Olfati-Saber and R.M. Murray. Consensus problems in network of agents with switching topology and time delays. *IEEE Trans. on Automatic Control*, 49(9), 2004.
- R. Olfati-Saber, A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- W. Ren. On consensus algorithms for double-integrator dynamics. In *IEEE Transactions on Automatic Control*, 53(6):1503–1509, 2008.
- W. Ren and Y. Cao. Convergence on sampled-data consensus algorithms for double-integrator dynamics. In *47th IEEE Conference on Decision and Control*, 2008.
- W. Ren, R. W. Beard, and E. M. Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference*, 2005.
- A. Seuret. A novel stability analysis of sampled-data systems with applications to multi-rate sampling and packet loss. *submitted to Automatica*, 2010.
- A. Seuret, C. Edwards, S.K. Spurgeon, and E. Fridman. Static output feedback sliding mode control design via an artificial stabilizing delay. *IEEE Trans. on Automatic Control*, 54(2):256–265, 2007.
- W. Ren and R. W. Beard. *Distributed Consensus in Multi-vehicle cooperative Control: theory and applications*. Springer, 2008.
- L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. In *42th IEEE Conference on Decision and Control*, 2003.