Improved double integrator consensus algorithms

Gabriel Rodrigues de Campos

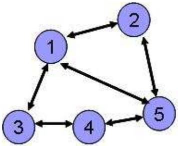
NeCS Team
CNRS - GIPSA-Lab Automatic Department
INRIA Rhône-Alpes

September 2010

Motivation

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.



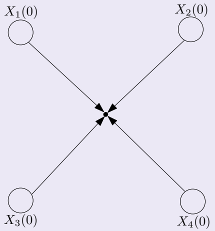
$$\Longleftrightarrow \begin{cases} \dot{x}_i = u_i \\ u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) \end{cases} \Longleftrightarrow \quad \dot{x} = -Lx$$

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

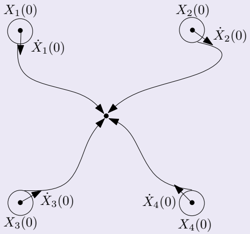
$$L = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & -1 & 0 & -1 & 3 \end{bmatrix}$$

Motivation

Simple Integrator agents dynamics:



Double Integrator agents dynamics:



The advantage of double integrator systems is that they fits to several robotics applications.

Motivation

Take the classical double integrator consensus algorithm

$$\ddot{x}(t) = u(t); \quad (1a)$$
$$u(t) = -\sigma\dot{x}(t) - Lx(t)\ , \quad (1b)$$

We will then have a position consensus taking into account initial velocities

Single Integrator alg. Vs Double Integrator alg.

Consensus Algorithms		
	Simple Int.	Double Int.
Symmetric Graphics	OK	OK
Asymmetric Graphics	OK	NO

Main Objective

Design an improved consensus algorithm for continuous-time multi-agent systems

Content

. Problem Statement
. Model definition
. Stability analysis
. Examples
. Conclusions

Based on:

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

Gabriel Rodrigues de Campos, Alexandre Seuret,
NecSys'10, France

Double Integrator Consensus

Consider the classical double integrator consensus algorithm

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

where x represents the vector containing the agents variables.
By introducing the augmented vector $y(t) = [x^T(t)\ \dot{x}^T(t)]^T$,

$$\dot{y}(t) = \begin{bmatrix} 0 & I \\ -L & -\sigma I \end{bmatrix} y(t) = \bar{L}y(t) \,. \tag{3}$$

Remark:

$\bar{L}$ has then positive eigenvalues if L is asymmetric, for $\forall\ \sigma$.

Delayed Consensus Algorithms

Main Idea

Introduce delays in the algorithm to improve the stability performances

Take for example an oscillating system defined by

$$\ddot{x}(t) + w_0^2 x(t) = u,$$

- Control law $u(t) = k_1 x(t) - k_2 \dot{x}(t)$ stabilize this system under an appropriate choice of k_1 and k_2.
- If velocity sensors are not available,then we can introduce the delayed control law:
$$\ddot{x}(t) + w_0^2 x(t) = k_1^* x(t) - k_2^* x(t-\tau).$$
Under some conditions on τ, the delayed component can been seen as
$$u(t) = \approx (k_1^* + k_2^*) x(t) + k_2^* \tau \dot{x}(t).$$

Delayed Consensus Algorithms

If we take $\sigma = 0$, the trivial double integrator algorithm can be expresses as:

$$\ddot{x}(t) = -Lx(t)\ , \tag{4}$$

and the previous algorithm is modified into a new algorithm defined by

$$\ddot{x}(t) = -(L + \delta^2 I)x(t) + \delta^2 x(t - \tau) \tag{5}$$

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

Remark:

Algorithm's stochastic proprieties remain intact $(+\delta^2 - \delta^2)$.

Delayed Consensus Algorithms

If we take $\sigma = 0$, the trivial double integrator algorithm can be expresses as:

$$\ddot{x}(t) = -Lx(t) \; , \tag{6}$$

and the previous algorithm is modified into a new algorithm defined by

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

Advantages:

- Reduces information quantity needed for control
- No more need of velocity sensors

Drawbacks:

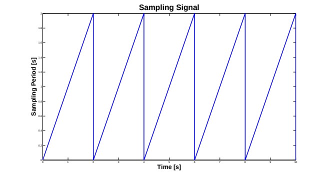
- Large memory is needed in order to keep x values between $[t - \tau, t]$

Delayed Consensus Algorithms

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.



Advantages: Smaller memory requirement
Drawbacks: More dedicated stability analysis

Finally the proposed algorithm is

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

where δ and T are now two additional control parameters.

Considering a performance optimisation:

We will then propose a method to choose appropriately the algorithm parameters δ and T for a given L

Exponential Stability :

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:

$$|x(t; t_0, \phi)| \leq F|\phi|_\tau e^{-\alpha(t-t_0)}. \tag{9}$$

Model Transformation

Let's take a change of coordinates $x = Wz$ such that

$$ULW = \begin{bmatrix} \Delta & \vec{0} \\ \vec{0}^T & 0 \end{bmatrix}, \tag{10}$$

where $\Delta \in \mathbb{R}^x$, 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 (8) can be rewritten using $z_1 \in \mathbb{R}^{N-1}$,

$z_2 \in \mathbb{R}$ and the matrix Δ is given in (10):

$$\ddot{z}_1(t) = -(\Delta + \delta^2 I)z_1(t) + \delta^2 z_1(t_k), \tag{11a}$$

$$\ddot{z}_2(t) = -\delta^2 z_2(t) + \delta^2 z_2(t_k), \tag{11b}$$

Previous Notation

Considering

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

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

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

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

Main Result

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

$$\Pi_1 + h_\alpha(T,0)M_2^T X M_2 + f_\alpha(T,0)\Pi_2 < 0, \tag{12}$$

$$\begin{bmatrix} \Pi_1 + h_\alpha(T,T)M_2^T X M_2 & g_\alpha(T,T)N \\ * & -g_\alpha(T,T)R \end{bmatrix} < 0, \tag{13}$$

where

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

$$M_0 = \begin{bmatrix} A(\delta) & A_d(\delta) \end{bmatrix}, M_1 = \begin{bmatrix} I & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 0 & I \end{bmatrix}, M_3 = \begin{bmatrix} I & -I \end{bmatrix}.$$

Then, the consensus algorithm is thus α_g-stable, where
$\alpha_g = \min\{\alpha, -\log(|\cos(\delta T)|)\}$.
Moreover the consensus equilibrium is given by
$x(\infty) = U_2(x(0) + \gamma_{\delta T}\dot{x}(0)),$
with $\gamma_{\delta T} = \sin(\delta T)/(\delta(1-\cos(\delta T))) = \tan((\pi - \delta T)/2)/\delta$

Sketch of the Proof:

Step 1)

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

$$\Downarrow \quad \textit{Model Transformation}$$

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

$$\ddot{z}_2(t) = -\delta^2 z_2(t) + \delta^2 z_2(t_k),$$

Sketch of the Proof:

Step 2) Exponential stability of z_1

Consider the following Functional:

$$\begin{aligned}\bar{V}(t,y_t) = & \, y^T(t)Py(t) + f_\alpha(T,\tau)\zeta_0^T(t)[S_1\zeta_0(t) + 2S_2y_k] \\ & + f_\alpha(T,\tau)\int_{t_k}^t \xi^T(s)M_0^T RM_0\xi(s)ds \\ & + (f_\alpha(T,0) - f_\alpha(T,\tau) - \tau/Tf_\alpha(T,0))y_k^T Xy_k\end{aligned}$$

with $y = [z_1^T(t)\ \dot{z}_1^T(t)]^T$.

If LMI's of the theorem are satisfied, then 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)

Sketch of the Proof:

Step 3) Stability of z_2

Integrating (11b), we obtain

$$\begin{bmatrix} z_2(t_{k+1}) \\ \dot{z}_2(t_{k+1}) \end{bmatrix} = \begin{bmatrix} 1 & \gamma_{\delta T}(1-cos(\delta T)^{k+2}) \\ 0 & cos(\delta T)^{k+1} \end{bmatrix} \begin{bmatrix} z_2(0) \\ \dot{z}_2(0) \end{bmatrix} \quad (14)$$

(14) is stable for any sampling period T and any δ such that $\delta T \neq 0\ [\pi]$, i.e $\delta T \neq k\pi$.

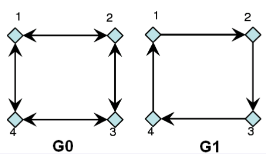
The variable z_2 converges to

$$z_2(\infty) = z_2(0) + \gamma_{\delta T}\dot{z}_2(0) = U_2\left(x(0) + \gamma_{\delta T}\dot{x}(0)\right) \quad (15)$$

where $\gamma_{\delta T} = \frac{\sin(\delta T)}{(\delta(1-\cos(\delta T)))} = \frac{\tan((\pi-\delta T)/2)}{\delta}$

Simulation Scenario

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



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 $\dot{x}^T(0) = [1\ 2\ 3\ 2]$.

Controller parameters optimization results

$$\max \alpha_g(\delta T)$$

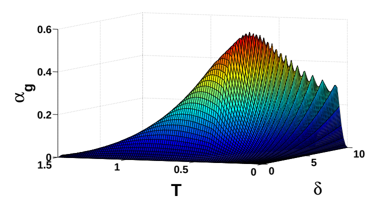


Figure: Exponential decay rate for G_0

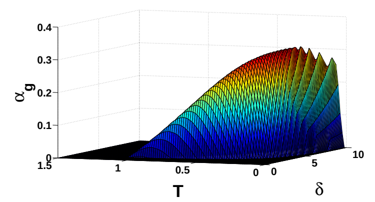
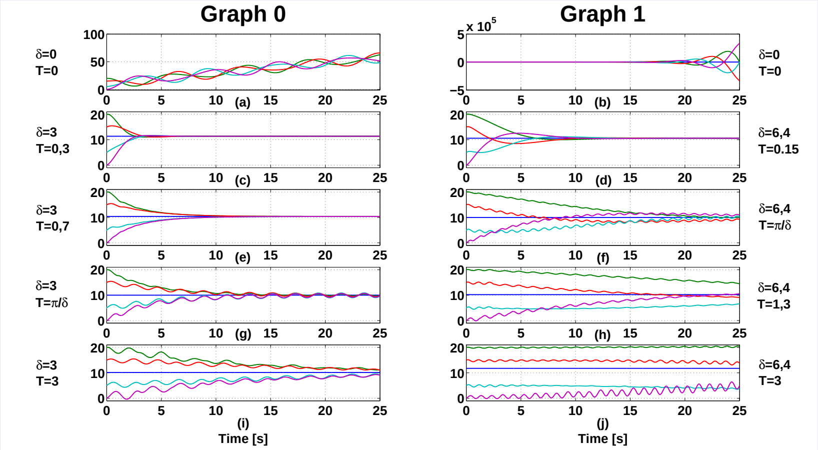


Figure: Exponential decay rate for G_1

Algorithm Convergence (Evolution of the agents state for several values of the sampling period T)



Conclusions and Perspectives

The proposed algorithm

. Reduces information quantity needed for control (if $\sigma = 0$)
. No more need of velocity sensors
. Economical, space and calculation savings
. Exponential stability of the solutions is achieved

Drawbacks

. The proposed stability criteria expressed in term of LMIs complexity will drastically increase for large networks.

Perspectives

. Development of new stability tools for a large agent network.

Thank you for your attention