



LUND UNIVERSITY

Robust synchronization of networks of heterogeneous double-integrators

Lovisari, Enrico; Carli, Ruggero

Published in:
Proceedings of the 51st Conference on Decision and Control

2012

[Link to publication](#)

Citation for published version (APA):
Lovisari, E., & Carli, R. (2012). Robust synchronization of networks of heterogeneous double-integrators. In *Proceedings of the 51st Conference on Decision and Control*

Total number of authors:
2

General rights

Unless other specific re-use rights are stated the following general rights apply:
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Robust synchronization of networks of heterogeneous double-integrators

Ruggero Carli

Enrico Lovisari

Abstract—This paper deals with the problem of clock synchronization in WSN. Clocks are modeled as non-identical double integrators with uncertain zeros and outputs corresponding to their time-readings. Synchronization is achieved through a control strategy based on second order linear consensus-like algorithm, and its robustness with the respect to the uncertainties on the clocks' dynamics is studied resorting to recently developed tools for synchronization of heterogeneous networks. An implementable, nonlinear version of the previous control strategy is also revised. Numerical simulations show that the behavior of the linear model which is analyzed here well approximates that of the implementable, realistic one.

I. INTRODUCTION

During the last decade there has been an increasing interest in Networked Control System for *synchronization* problems. The usual scenario is that of large networks of identical anonymous agents, which have access to some partial information, and aim at agreeing on some quantity of interest, possibly time-varying. A common feature of these problems is the fact that there is a fundamental constraint on the information flow: data are distributed among a large number of nodes communicating with each other only through some communication network which typically allows each agent to receive information only from a small subset of other agents.

The most studied synchronization algorithm in the literature is the linear consensus in which the agents need to agree on a static quantity. In this algorithm the agents behave as simple integrators whose input is a linear combination of the outputs of the neighbors. This model well describes many cases of interest, for example when consensus is used as a tool for formation control [1], [2], [3], distributed estimation [4], sensor and camera calibration [5], [6], and distributed optimization [7], [8].

However, there are many other situations in which agents still need to agree on some quantity, but their nature is too complex to be modeled as simple integrators. Examples can be found in formation control problems for robotic networks, where usually the dynamics are modeled as second order systems. Recently, many contributions have been provided on the synchronization of systems with high-order dynamics. In [2] the authors consider a network of equal agents, modeled as a generic SISO transfer function $N_0(s)$, where the inputs of the agents are computed as a linear combination of the

outputs of the neighbors, as in the classic linear consensus. Also in [9] the agents are assumed to have identical dynamics described by a general state-space representation (A, B, C) and the goal is to synchronize, entry-wise, the states of the agents. In [10] the agents are noisy integrators and inputs are filtered versions of the outputs, instead of being simple linear combinations. Finally, in [11] the authors propose a graphical criterion to check the synchronization of a generic heterogeneous network, namely, a network comprising agents with different dynamics.

Synchronization is also a key problem in wireless sensor networks (WSNs). This is mainly motivated by the fact that in many applications involving WSNs, like surveillance, targeting systems, monitoring areas, intrusion detection, it is essential that the nodes act in a coordinated fashion requiring global clock synchronization, that is, all the nodes need to refer to a common notion of time. The synchronization problem has been addressed via hierarchical algorithms such as leader election in a spanning tree [12], [13], or clustering of the network [14]. Despite the effectiveness of such algorithms, they suffer from poor scalability and are very sensitive to network topology changes.

Recently distributed architecture for clocks synchronization have been proposed, where there are no reference nodes and all nodes run exactly the same algorithm. These protocols have the advantage of being highly scalable and very robust to node failure and new node appearance. Fully distributed algorithms able to compensate for both clock rates and offsets are the Distributed Time-Sync Protocol in [15] and the Average Time-Sync Protocol in [16]. The first one is based on the cascade of two distributed least-squared algorithms, while the second one is based on the cascade of two first order consensus algorithms. They both are proved to synchronize a network of clocks in the absence of noise and time-delays. However, these strategies are highly non-linear and do not lead to a simple characterization of the effects noise on the steady-state performance.

Differently, [17] proposes a linear synchronization algorithm that allows for a formal analysis not only in the noiseless scenario in terms of rate of convergence, but also in a noisy setting in terms of the steady-state synchronization error. In [17] the local clocks are modeled as non-identical double integrators and the control strategy proposed is based on a second order linear consensus algorithm. The authors provide conditions guaranteeing synchronization under the assumption of no-communication noise and no-delays and provided that the rates of all the double integrators are small enough perturbations of a nominal value. However

The research leading to these results has received funding from the European Community's Seventh Framework Programme under agreement n. 257462 HYCON2 Network of excellence.

R. Carli and E. Lovisari are with the Department of Information Engineering, Università di Padova, Via Gradenigo 6/a, 35131 Padova, Italy e-mail:{carlirug,lovisari}@dei.unipd.it.

how big these perturbations could be, it is not addressed. Moreover it is important to remark that the protocol in [17] is proposed in a synchronous implementation, namely, all nodes are assumed to communicate at the same time instants. Clearly this assumption is unrealistic for implementing any clock synchronization algorithm over WSNs since nodes data transmissions and algorithm updating steps cannot occur exactly at the same time. This issue is overcome in [18] where a realistic implementation of the control law introduced in [17] is proposed in the context of WSNs applications. The authors refer to this realistic implementation as to the *pseudo synchronous* algorithm which is proved to be a perturbation of the synchronous version proposed in [17]. The *pseudo synchronous* algorithm is shown to be provably convergent even if also in this case the convergence result is only local, that is, the clocks' rates must be small perturbations of a nominal rate value. It is worth mentioning that a completely asynchronous version of the synchronization algorithm introduced in [17] has been proposed in [19]. The effectiveness of this asynchronous algorithm has been shown only through simulations and no theoretical results have been provided so far.

The contributions of this paper are twofold. Firstly in Section II we review the synchronization problem over a network of non-identical double integrators as posed in [17] focusing on the robustness issues which arise since the rates of the double integrators are, in general, different one from each other. In section III, we provide interesting insights on these issues. In particular resorting to robust control tools recently developed, we obtain novel conditions guaranteeing the synchronization which significantly extend the ones stated in [17] and which allow for an interesting characterization of the robustness properties of the control strategy proposed in [17]. Secondly, in Section IV we review the *pseudo-synchronous* algorithm proposed in [18]. Through numerical simulations we show that the *pseudo-synchronous* algorithm exhibits robustness properties similar to the ones obtained in Section III for synchronous version. This fact strengthens the local result provided in [18] for the convergence of the *pseudo-synchronous*.

In Section V we gather our conclusions while in Appendix A we shortly review the robust control tools proposed in [11].

A. Mathematical preliminaries

Before proceeding, we collect some useful definitions and notations. In this paper, $\mathcal{G} = (V, \mathcal{E})$ denotes an undirected graph where $V = \{1, \dots, N\}$ is the set of vertices and \mathcal{E} is the set of edges, i.e., $\mathcal{E} \subseteq V \times V$. Since \mathcal{G} is undirected, if $(i, j) \in \mathcal{E}$ then also $(j, i) \in \mathcal{E}$. A path in \mathcal{G} consists of a sequence of vertices (i_1, i_2, \dots, i_r) such that $(i_h, i_{h+1}) \in \mathcal{E}$ for every $h \in \{1, \dots, r-1\}$. A graph \mathcal{G} is connected if for any pair of vertices (i, j) there exists a path connecting i to j . Given a matrix $M \in \mathbb{R}^{N \times N}$, we define the induced graph \mathcal{G}_M by taking N nodes and putting an edge (j, i) in \mathcal{E} if $M_{ij} \neq 0$. Given a graph \mathcal{G} on V , the matrix M is compatible with \mathcal{G} if $\mathcal{G}_M \subseteq \mathcal{G}$. Given the node i , by \mathcal{N}_i we denote the set of its neighbors, i.e., $\mathcal{N}_i = \{j \in V | (i, j) \in \mathcal{E}\}$.

II. PROBLEM FORMULATION

Assume we have N units and that each unit i has a clock which is an oscillator able to periodically increment a counter by one unit, commonly known as tick. We assume that the periods Δ_i of these oscillators are unknown, but are "perturbed" values of a "nominal" and known period Δ . Therefore, the value of the i -th counter is $s_i(t) = \lfloor \frac{t-t_{0i}}{\Delta_i} \rfloor$, where $\lfloor a \rfloor$ indicates the floor of a , i.e., the largest integer smaller than or equal to a , and where t_{0i} is the time when the clock has been started. The units use these ticks to estimate time. Since only the nominal clock period Δ is known by the clocks, a natural time estimate is given by

$$y_i(t) = \Delta s_i(t) + y_i(t_{0i}) \quad (1)$$

where $y_i(t_{0i})$ is the initial offset which is an estimate of t_{0i} . Since the Δ_i 's are all different, if the units can not communicate with each other, then each clock will drift away from the others even under the ideal situation in which they are all initially synchronized, i.e., $y_i(0) = y_j(0)$ for all i, j . Formally, if

$$e_i(t) := y_i(t) - \frac{1}{N} \sum_{i=1}^N y_i(t), \quad i \in \{1, \dots, N\},$$

denote the synchronization error of the i -th unit, then, in absence of communication between the units, $\lim_{t \rightarrow \infty} \|e(t)\|^2 = \infty$, where $e(t) = [e_1(t), \dots, e_N(t)]^T$.

Now let $T \in \mathbb{R}$ be a sampling time. Observe that at the time instants hT , where $h = 0, 1, \dots$, the evolution of $y_i(hT)$ in (1) can be iteratively described as

$$\begin{aligned} x_i((h+1)T) &= \begin{bmatrix} 1 & \Delta \delta_i(h) \\ 0 & 1 \end{bmatrix} x_i(hT), \quad x_i(0) = \begin{bmatrix} y_i(0) \\ 1 \end{bmatrix} \\ y_i(hT) &= [1 \quad 0] x_i(hT) \end{aligned}$$

where $x_i(hT) \in \mathbb{R}^2$ and where $\delta_i(h) := s_i((h+1)T) - s_i(hT)$ gives the number of ticks in the interval $(hT, (h+1)T]$. If x'_i and x''_i denote the two components of x_i , then x'_i gives the time estimate, while $\Delta x''_i$ gives the oscillator period estimate. Notice that $s_i((h+1)T) - s_i(hT) = T/\Delta_i + \epsilon(hT)$ where $-1 < \epsilon(hT) < 1$ and so $\epsilon(hT)$ can be neglected if $T/\Delta_i \gg 1$ which will be assumed in the sequel. Accordingly the above recursion can be rewritten as

$$\begin{aligned} x_i((h+1)T) &= \begin{bmatrix} 1 & \frac{\Delta}{\Delta_i} T \\ 0 & 1 \end{bmatrix} x_i(hT), \quad x_i(0) = \begin{bmatrix} y_i(0) \\ 1 \end{bmatrix} \\ y_i(hT) &= [1 \quad 0] x_i(hT). \end{aligned} \quad (2)$$

Observe that the previous system corresponds to the output of a second order integrator with unknown parameters (in particular, one unknown zero), since $\{\Delta_i\}_{i=1}^N$ are not known.

Assume now that at the time instants hT , the nodes can exchange their local time $y_i(t)$ according to a graph \mathcal{G} having $\{1, \dots, N\}$ as set of vertices and in which there is an edge from j to i whenever node j can send information to node i . This information can be used by the nodes to correct their clock estimate $y_i(t)$ so that eventually all nodes will be synchronized, i.e., $y_i(t) \simeq y_j(t)$ for all i, j . A natural

approach to achieve synchronization is to control the nominal clock period Δ and the clock offset $y_i(0)$. Based on the information received from the neighboring nodes, we assume that each node can add a control $u_i(hT)$ to its state, namely,

$$x_i(hT^+) = x_i(hT) + u_i(hT) \quad (3)$$

where hT^+ denotes the time instant just after hT . It follows that system (2) is modified as

$$\begin{aligned} x_i((h+1)T) &= \begin{bmatrix} 1 & \frac{\Delta}{\Delta_i} T \\ 0 & 1 \end{bmatrix} x_i(hT^+) \\ &= \begin{bmatrix} 1 & \frac{\Delta}{\Delta_i} T \\ 0 & 1 \end{bmatrix} (x_i(hT) + u_i(hT)) \end{aligned} \quad (4)$$

and

$$y_i(h) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(hT), \quad (5)$$

From now on, for simplicity we will denote Δ/Δ_i by d_i .

The authors in [17] and [18] proposed the following linear control law

$$u_i(hT) = - \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sum_{j=1}^N k_{ij}(h) y_j(hT) \quad (6)$$

where f_1 and f_2 are two real parameters and where $k_{ij}(h)$ is the $i-j$ entry of the matrix $K(h) \in \mathbb{R}^{N \times N}$. Notice that at time hT the protocol requires the transmission of the output $y_j(h)$ from the node j to the node i if and only if $k_{ij}(h) \neq 0$.

As in [17] and [18] we assume the following properties.

Assumption II.1 *The control law in (6) is time-invariant, namely, $K(h) = K$ for all $h \in \mathbb{N}$, where*

- (i) *K is symmetric and positive semidefinite;*
- (ii) *$K\mathbf{1} = 0$ being $\mathbf{1}$ the N -dimensional column vector with all the components equal to 1 and the algebraic multiplicity of the eigenvalue 0 is one¹.*

In subsection II-A we show that, given a connected undirected graph \mathcal{G} , it is always possible to build in a distributed way a matrix K compatible with \mathcal{G} and satisfying the above properties (i) and (ii). Introducing the $2N$ dimensional vector $x(hT)$ having $x'(hT) = [x'_1(hT), \dots, x'_N(hT)]^T$ as the first N components and $x''(hT) = [x''_1(hT), \dots, x''_N(hT)]^T$ as the second N components, the N equations in (4) and in (6) can be collected in

$$x((h+1)T) = A(f_1, f_2, K, D) x(hT) \quad (7)$$

where

$$A(f_1, f_2, D, K) = \begin{bmatrix} I & DT \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_1 K & 0 \\ f_2 K & 0 \end{bmatrix} \right)$$

being D the diagonal matrix having d_1, \dots, d_N as diagonal elements, i.e., $D = \text{diag}\{d_1, \dots, d_N\}$. In case the matrix $A(f_1, f_2, K, D)$ yields all the components of $x'(hT)$ follow

¹It is worth remarking that the condition $K\mathbf{1} = 0$ is imposed in order to avoid useless input in case the clocks are already synchronized.

asymptotically the same ramp function, that is, there exist $\alpha \in \mathbb{R}_{>0}$ and $\beta \in \mathbb{R}$ such that

$$\lim_{h \rightarrow \infty} (x'(h) - (\alpha hT + \beta)\mathbf{1}) = 0,$$

we say that this matrix yields the *synchronization*, as all the entries of $x'(h)$ are steered to the same time-varying value.

The next proposition, whose proof can be found in [18], characterizes some convergence properties of system in (7). Before stating it, we introduce the following notational definition: let $\lambda_1, \dots, \lambda_N$ denote the N eigenvalues of K and, without loss of generality, assume that

$$0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_N. \quad (8)$$

Proposition II.1 *Consider system (7) under Assumption II.1. Moreover assume that $D = I$, i.e., $d_i = 1$ for all $i \in \{1, \dots, N\}$. Then, the matrix $A(f_1, f_2, K, I)$ yields the synchronization if and only if the following conditions are satisfied*

$$\begin{aligned} f_1 &> 0, & f_2 &> 0, \\ 0 < \lambda_N &< \frac{4}{2f_1 + Tf_2}. \end{aligned} \quad (9)$$

The previous proposition is valid for $D = I$. However, since the eigenvalues of A depend continuously on the matrix D we have that conditions in (9) still ensure the synchronization even in the case where D is a small enough perturbation of the identity matrix I .

In this paper we aim at providing a more refined analysis of the robustness of the control law in (6) with respect to the matrix D . Specifically, let us denote the matrix D as the sum of the identity matrix I and of the diagonal perturbation matrix $\tilde{D} = \text{diag}\{\tilde{d}_1, \dots, \tilde{d}_N\}$, i.e., $D = I + \tilde{D}$, where $\tilde{d}_i > -1$ for all $i \in \{1, \dots, N\}$. Then, the problem of interest is : Given a triple (f_1, f_2, K) , satisfying conditions in (11), characterize the set of perturbation matrices

$$\begin{aligned} \mathcal{P}(f_1, f_2, K) &:= \\ &\{D \text{ s.t. } A(f_1, f_2, K, D) \text{ yields the synchronization}\}. \end{aligned}$$

In section III we will provide some interesting insights on the above problem.

A. Metropolis weights

Assume that $D = I$. Then, given a connected undirected graph \mathcal{G} , the problem is to determine a matrix K compatible with \mathcal{G} and the parameters f_1, f_2 such that conditions in (9) are satisfied. In [18] it is proposed the following K

$$K_{ij} = \begin{cases} -\frac{1}{\max\{v_i, v_j\}} & \text{if } (i, j) \in \mathcal{E} \text{ and } i \neq j \\ -\sum_{j \neq i} K_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

where v_i is the degree of the node i , namely $v_i = |N_i|$, and the constants

$$f_1 = 1/2 \quad \text{and} \quad f_2 = 1/(2T).$$

The weights K_{ij} are called in literature Metropolis weights. From now on, we will denote a matrix K built as in (10), by K_{Metr} . It is well known that, if the undirected graph \mathcal{G} is connected then the associated matrix K_{Metr} yields $0 < \lambda_i < 2$, $i \in \{2, \dots, N\}$. Then it is straightforward to verify that the triple $f_1 = 1/2$, $f_2 = 1/(2T)$, K_{Metr} satisfies the conditions in (9). An additional goal of the next section will be to provide some characterizations of the set

$$\mathcal{P}(f_1 = 1/2, f_2 = 1/(2T), K_{Metr}).$$

It is worth saying that in this paper we focus on the Metropolis weights as special case since, given the connected graph \mathcal{G} , the associated matrix K_{Metr} can be easily built in a completely distributed way. Indeed what a node needs in computing the weights K_{ij} is only the knowledge of the degrees of its neighboring nodes in the graph \mathcal{G} . This fact makes the use of the Metropolis weights quite wide in wireless sensor networks applications.

III. ROBUSTNESS ANALYSIS OF THE SYNCHRONIZATION PROTOCOL

In order to perform a robustness analysis of the iterative algorithm in (7), we will resort to the robust control tools recently developed in [11]. From a non-standard application of these tools we get the following result which extends the conditions in (9) to the case of heterogeneous double integrators.

Theorem III.1 *Consider system (7) under Assumption II.1. If the following conditions are satisfied*

$$\begin{aligned} f_1 &> 0, & f_2 &> 0, \\ 0 &< \lambda_N < \frac{4}{2f_1 + Tf_2}, \\ 0 &< \lambda_N < \frac{4}{2f_1 + Tf_2 d_i}, & i &\in \{1, \dots, N\}, \end{aligned} \quad (11)$$

then the matrix $A(f_1, f_2, K, D)$ yields the synchronization.

Some observations are now in order. Firstly observe that, in order to deal with the fact that the values $\{d_i\}_{i=1}^N$ might be, in general, different from each other, Theorem III.1 requires an extra condition with the respect to the ones assumed in Proposition II.1. This condition relates the values $\{d_i\}_{i=1}^N$ to the parameters f_1 , f_2 and to the eigenvalue λ_N . Secondly, the result stated in Theorem III.1 is in some sense weaker than the one stated in Proposition II.1. Indeed, while conditions in (9) are necessary and sufficient to synchronization, conditions in (11) are only sufficient. Finally let the triple (f_1, f_2, K) be given and assume that f_1 , f_2 and λ_N satisfy conditions in (11), and, in turn, also the first two conditions in (11). From the above Theorem it follows that, if

$$0 < d_i < \frac{4 - 2f_1\lambda_N}{Tf_2\lambda_N},$$

then the matrix $A(f_1, f_2, K, D)$ yields the synchronization. Hence if we introduce the set of diagonal matrices

$$\bar{\mathcal{P}}(f_1, f_2, K) := \left\{ D : 0 < d_i < \frac{4 - 2f_1\lambda_N}{Tf_2\lambda_N}, \forall i \right\}$$

we have that

$$\bar{\mathcal{P}}(f_1, f_2, K) \subseteq \mathcal{P}(f_1, f_2, K).$$

It is interesting to notice that the condition $\lambda_N < \frac{4}{2f_1 + Tf_2}$ implies $\frac{4 - 2f_1\lambda_N}{Tf_2\lambda_N} > 1$.

We conclude this section specializing the result of Theorem III.1 to the case of Metropolis weights.

Corollary III.2 *Let \mathcal{G} be a undirected connected graph and let K_{Metr} be the matrix associated to \mathcal{G} built as in (10). Consider the matrix $A(f_1, f_2, K_{Metr}, D)$ where $f_1 = 1/2$ and $f_2 = 1/(2T)$ and where the matrix D is such that*

$$0 < d_i < 2, \text{ for all } i \in \{1, \dots, N\}.$$

Then the matrix $A(f_1, f_2, K_{Metr}, D)$ yields the synchronization.

Remark III.3 *The result stated in the above Corollary is quite remarkable. Indeed, if $0 < d_i < 2$ for all $i \in \{1, \dots, N\}$, then the synchronization is guaranteed independently from the number of agents and from the topology of the communication graph \mathcal{G} . The only requirement is the graph \mathcal{G} to be connected.*

Remark III.4 *Observe that the condition $0 < d_i < 2$ can be expressed in terms of the oscillator period Δ_i as $\frac{\Delta_i}{2} < \Delta_i < +\infty$.*

Remark III.5 *We tried to analyze the robustness of the control law in (6) also through a standard application of the small gain Theorem. The results we found were weaker than the ones illustrated in this section. To be more precise consider the circle graph, namely, the graph containing the edges $(i, i+1)$, $i \in \{1, \dots, N-1\}$, and $(1, N)$. Let $d_i := 1 + \tilde{d}_i$. Then the result we obtained applying the small gain Theorem was that the matrix $A(f_1, f_2, K_{Metr}, D)$ is synchronizing if $|\tilde{d}_i| < \gamma(N)$ where the bound $\gamma(N)$ is such that $\lim_{N \rightarrow \infty} \gamma(N) = 0$. In other words the size of the robustness interval obtained through the small gain Theorem goes to zero as N tends to 0. We found similar results for other families of Cayley graphs, like the d -dimensional toruses for any arbitrary value of the dimension d .*

IV. ROBUST CLOCKS SYNCHRONIZATION IN WIRELESS SENSORS NETWORK (WSNs)

Observe that the control strategy reviewed in Section II is proposed in a synchronous implementation, namely, all the nodes are assumed to perform their transmitting and updating actions exactly at the same time instants. Clearly this assumption is unrealistic for implementing any clock synchronization algorithm over WSNs since nodes transmission and algorithm updating steps cannot occur exactly at the same time. Indeed, in a fully distributed synchronization algorithm, these operations can be performed by the nodes relying only on their local time estimates and, in absence of synchronization, these estimates differ from each other.

In [18], a realistic implementation of the approach illustrated in Section II is proposed in the context of WSNs applications. In particular the authors of [18] introduced suitable modifications to the iterative algorithm in (7) that allow the nodes to overcome the fact that, in a realistic setup, they are unable to carry out their transmission and updating actions synchronously. They refer to this realistic implementation as to the *pseudo-synchronous* algorithm showing that it can be described as a particular non-linear perturbation of the synchronous version in (7). For completeness, we next review the *pseudo-synchronous* algorithm. To do so we first need to specify how the nodes select the transmission and the updating time instants. Let us introduce the following notational definitions.

- By $t_{tx,h,i}$ we denote the time instant in which the i -th node performs its h -th transmission; here we assume a broadcast model, i.e., each node transmits, at the same time, the same information to all its neighbors;
- By $t_{up,h,i}$ we denote the time instant in which the i -th node performs its h -th updating step.

As previously observed, $t_{tx,h,i}$ and $t_{up,h,i}$ can be determined by the node i relying only on its local information. We define:

- $t_{tx,h,i}$ as the first time such that $x'_i(t_{tx,h,i}) = hT$, namely the node i transmits when its time estimate is equal to hT for the first time;
- $t_{up,h,i}$ to be equal to

$$\max \{t_{tx,h,j} | j \in \mathcal{N}_i \cup \{i\}\},$$

namely the node i updates its state only after having received the information from all its neighbors and after having performed its transmission.

The nodes transmission and updating steps of the *pseudo-synchronous* algorithm can be formally described as follows.

According to the *pseudo-synchronous* algorithm, during the h -th iteration each node i performs the following actions:

- At instant time $t_{tx,i}(h)$ the node i broadcasts to its neighbors the state $x_i(t_{tx,i}(h))$.
- For all $j \in \mathcal{N}_i$, at time $t_{tx,j}(h)$ the node i receives from the node j the value $x_j(t_{tx,j}(h))$, it computes the difference $x_j(t_{tx,j}(h)) - x_i(t_{tx,j}(h))$ and it stores it in the memory.
- At time $t_{up,i}(h)$ the node i updates its state using all the stored data as follows

$$x_i(t_{up,h,i}^+) = x_i(t_{up,h,i}) + u_i(t_{up,h,i})$$

where

$$u_i(t_{up,h,i}) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \sum_{j \in \mathcal{N}_i} k_{ij} (x'_j(t_{tx,h,j}) - x'_i(t_{tx,h,j})).$$

Remark IV.1 Observe that it might happen that $x'_i(t_{up,i}(h)) < (h+1)T$ and that $x'_i(t_{up,i}(h)^+) > (h+1)T$ so that there would not be any time t in which $x'_i(t) = (h+1)T$ and, in turn, no way to determine $t_{tx,i}(h+1)$. If this situation occurs, then it is imposed that $t_{tx,i}(h+1) = t_{up,i}(h)$ and that the node i

transmits its state $x_i(t_{up,i}(h)^+)$ where it is assumed that $x'_i(t_{up,i}(h)^+) = (h+1)T$.

As mentioned, the *pseudo-synchronous* algorithm can be written as a perturbation of the synchronous algorithm (7). This fact is formally shown by introducing a set of suitable sampling instants t_h , $h \in \mathbb{N}$, defined as

$$t_h = \min_i \{t_{tx,h,i}\},$$

namely, t_h represents the first time instant in which the estimate of a node reaches the value hT . Additionally, let

$$x(h) := x(t_h).$$

In [18], it is shown that

$$x(h+1) = A(f_1, f_2, K, D)x(h) + \Phi(x(h), \delta T_h, \delta t_{up,h}, \delta t_{tx,h})$$

where the function Φ depends continuously on the quantities $x(h)$, δT_h , $\delta t_{up,h}$, $\delta t_{tx,h}$ being

$$\begin{aligned} \delta T_h &:= t_{h+1} - t_h - T \\ \delta t_{up,h} &:= [t_{up,h,1} - t_h, \dots, t_{up,h,N} - t_h]^T \\ \delta t_{tx,h} &:= [t_{tx,h,1} - t_h, \dots, t_{tx,h,N} - t_h]^T. \end{aligned}$$

Observe that $\delta t_{up,h}$ and $\delta t_{tx,h}$ accounts for the lack of synchronicity in performing the updating and transmitting actions, while $t_{h+1} - t_h - T$ measures the length of the h -th sample period with the respect to the nominal length T .

Through arguments related to the Center Manifold Theorem, the *pseudo-synchronous* algorithm is shown to be provably convergent, see [18]. However this is only a local convergence result, in the sense that what it is proved is only the existence of $\epsilon > 0$ and of a neighborhood $W \in \mathbb{R}^N$ of the origin such that the synchronization is asymptotically reached if $x'(0) \in W$ and $|1 - d_i| < \epsilon$ for all $i \in \{1, \dots, N\}$. How big ϵ and the neighborhood W could be, it is not quantified in [18]. However, since the *pseudo-synchronous* can be seen as a perturbation of the synchronous algorithm (7), one could expect that some of the robustness properties illustrated in Section III for (7), might be inherited also by the *pseudo-synchronous* algorithm. We try to verify this fact through numerical simulations. Specifically, we run a number of simulations varying the size of the network N , the value of the sampling time T and the communication topology \mathcal{G} . Moreover in most of the simulations we assumed that

- the graph \mathcal{G} is connected and in the control law is implemented with $f_1 = 1/2$, $f_2 = 1/(2T)$ and $K = K_{Metr}$;
- for $i \in \{1, \dots, N\}$, the initial condition $x'_i(0)$, is randomly generated within the interval $[0, T/10]$; and
- for $i \in \{1, \dots, N\}$, the value of d_i is randomly chosen within the interval $(1/2, 3/2)$.

Under the above assumptions, the *pseudo-synchronous* algorithm always attained the asymptotic synchronization. Similarly to result stated in Corollary III.2, we can argue the

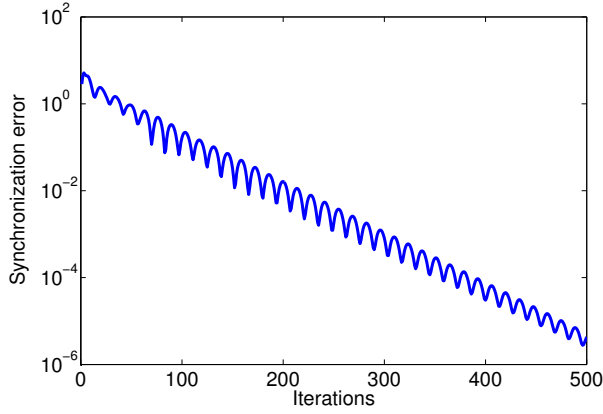


Fig. 1. Trajectories of the synchronization error generated by the *pseudo-synchronous* algorithm.

existence also for the *pseudo-synchronous* algorithm of a robustness interval which is independent from the number of clocks N and the communication graph \mathcal{G} .

In Figure 1 we plot an example of the behavior of $\log N^{-1/2} \|e(h)\|$ where $e(h) = [e_1(h), \dots, e_N(h)]^T$ being $e_i(h) = x'_i(h) - \sum_{j=1}^N x'_j(h)$. In this example we consider a connected random geometric graph G with $N = 100$ points uniformly distributed in the unit square, and placing an edge between each pair of points at distance less than 0.2. We associate to \mathcal{G} the matrix K_{Metr} and we set $T = 100$. We assume that $x'_i(0)$ and d_i are uniformly chosen, respectively, in $[0, 10]$ and in $(1/2, 3/2)$. The plot reported is the result of the average over 1000 Monte Carlo runs, randomized with respect to both the graph and the initial conditions. Observe that the trajectory converge exponentially to zero.

Remark IV.2 We simulated the *pseudo-synchronous algorithm* also for values of d_i uniformly chosen within $(0, 2)$. In some of the realizations generated, we obtained that $\lim_{h \rightarrow \infty} \|e(h)\| = \infty$.

V. CONCLUSIONS

This paper provides an analysis of synchronization robustness for networks of different double integrators, which models a network of clocks. It is shown that synchronization is achieved if conditions involving only the spectrum of the interconnection matrix and the rate of the clocks are satisfied. A realistic model for clocks is also revised, which is a nonlinear perturbation of the one studied here. Numerical simulations show good accordance among the two models.

REFERENCES

- [1] J. Cortes, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, aug. 2006.
- [2] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transaction on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, sep. 2004.
- [3] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri, "Communication constraints in the average consensus problem," *Automatica*, vol. 44, no. 3, pp. 671–684, 2008.
- [4] R. Olfati-Saber, "Distributed kalman filter with embedded consensus filters," in *Proceedings of 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05.*, dec. 2005, pp. 8179 – 8184.
- [5] A. Fagiolini, S. Martini, and A. Bicchi, "Set-valued consensus for distributed clock synchronization," in *Proceedings of IEEE Conference on Automation Science and Engineering. CASE '09*, august 2009, pp. 116–121.
- [6] D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri, "Autonomous calibration algorithms for networks of cameras," in *American Control Conference*, 2012.
- [7] A. Nedic, A. Ozdaglar, and P. Parrilo, "Constrained consensus and optimization in multi-agent networks," *Automatic Control, IEEE Transactions on*, vol. 55, no. 4, pp. 922–938, april 2010.
- [8] B. Johansson, M. Rabi, and M. Johansson, "A randomized incremental subgradient method for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1157–1170, 2009.
- [9] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557 – 2562, 2009.
- [10] I. Lestas and G. Vinnicombe, "Heterogeneity and scalability in group agreement protocols: Beyond small gain and passivity approaches," *Automatica*, vol. 46, no. 7, 2010.
- [11] E. Lovisari and U. Jönsson, "A nyquist criterion for synchronization in networks of heterogeneous linear systems," in *Proceedings of 2th IFAC Workshop on Distributed Estimation and Control in Networked Systems. Necsyst'10*, 2010.
- [12] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timingsync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded networked sensor systems. (SenSys'03)*, 2003, pp. 138–149.
- [13] M. Maròti, B. Kusz, G. Simon, and Àkos Ldeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, 2004, pp. 39–49.
- [14] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, 2002, pp. 147–163.
- [15] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *45th IEEE Conference on Decision and Control (CDC'06)*, San Diego, December 2006, pp. 2734–2739.
- [16] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.
- [17] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Optimal synchronization for networks of noisy double integrators," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1146–1152, 2011.
- [18] R. Carli and S. Zampieri, "Networked clock synchronization based on second order linear consensus algorithms," in *49th IEEE Conference on Decision and Control (CDC'10)*, Atlanta, December 2010.
- [19] R. Carli, E. D'Elia, and S. Zampieri, "A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks," in *Proceedings of the 50th IEEE Conference on Decision and Control. CDC'11*, 2011.