Networked clock synchronization based on second order linear consensus algorithms

Ruggero Carli Sandro Zampieri

Abstract— **In this paper a distributed algorithm for clock synchronization is proposed. This algorithm is based on an extension of the consensus algorithm able to synchronize a family of double integrators. Since the various clocks may have different drifts, the algorithm needs to be designed so that it can work also in case of heterogeneous double integrators. Through a robust control analysis it is possible to determine the maximum admissible level of heterogeneity yielding synchronization. The first part of the paper is devoted to the analysis of an unrealistic synchronous implementation of the algorithm. However, in the last part of the paper we propose a realistic pseudo-synchronous implementation which is proved to be a perturbation of the synchronous one. From arguments related the center manifold theorem, the stability of the pseudo-synchronous is finally proved.**

I. Introduction

One of the key problems in sensor networks is time-synchronization. Sensor networks are used in a large number of applications which cover a wide range of fields, such as, surveillance, targeting systems, controls, communications, monitoring areas, intrusion detection, vehicle tracking and mapping. In many of these applications it is essential that the nodes act in a coordinated and synchronized fashion requiring global clock synchronization, that is, all the nodes of the network need to refer to a common notion of time.

A common approach to solve the synchronization problem is to create a hierarchical structure within the network. The strategy proposed in [1], [2] consists in electing a reference node and creating a spanning tree rooted at this reference node, where each children synchronizes itself with respect to its parent. In [3], the network is divided into distinct clusters, each with an elected cluster-head. All nodes within the same cluster synchronize themselves with the corresponding cluster-head, and each cluster-head synchronizes itself with a another cluster-head. Although these two strategies have been experimentally tested showing remarkable performance, they suffer from robustness and scalability issues. For instance, if a node dies or a new node is added, then it is necessary to rebuild the tree or the clusters, at the price of additional implementation overhead and possibly long periods in which the network or part of it is poorly synchronized.

Recently fully distributed algorithms for clocks synchronization have appeared. The authors in [4] introduced a protocol able to compensate for different clock offsets but not for different clock skews. On the opposite, the algorithm proposed in [5] compensates for the clock skews but not for the offsets. Distributed protocols that can compensate for both clock skews and offsets have been proposed in [6], [7]. Of note is the fact that both these strategies are highly non-linear and do not lead to a simple characterization of the effects of noise on the steady-state performance.

In this work we present a novel distributed synchronization protocol which is based on a second order linear consensus algorithm. The term consensus refers to a general class of distributed algorithms that allow multiple agents to converge to the same quantity of interest using only local communication, see e.g. [8]. The advantage of using a simple linear feedback strategy allows for the analysis of the performance also in the presence of measurement and process noise.

Time synchronization of clocks with different speeds provides an interesting class of systems. In fact each local clock can be modeled as the output of a double integrator whose rate is not perfectly known. Moreover this rate is slightly different from one clock to another, therefore even if all clocks are perfectly synchronized at one time, they will slowly diverge from each other if no compensation or resynchronization is applied. During the last years, synchronization of networked higher order systems has received a large interest. Most of the available results are for synchronization of either non-identical systems which are strictly stable [9], or identical linear systems [10], [11]. Specific attention has been given to the synchronization of double integrators, which are unstable systems with a ramp mode [12]. However the strategy in [12], as the strategies in [10], [11], strongly relies on the assumption that all systems are identical.

In this paper we propose a novel distributed clock synchronization protocol based on a consensus algorithm for non identical double integrators whose rates of growth are not known nor measurable. In the first part of the paper we formally analyze our technique in a unrealistic synchronous implementation, i.e., all the nodes are supposed to communicate at the same time instant. We provide convergence guarantees on the protocol parameters and we perform a robustness analysis to evaluate the admissible uncertainties of the clocks speeds with the respect to a pre-assigned nominal value, while maintaining the convergence properties. In the second part of the paper we present a more realistic pseudo-synchronous implementation which can be formally modeled as a perturbation of the synchronous version. This pseudo-synchronous implementation is provably convergent and, through simulations, we evaluate its performance also in the case where the communication delays are not negligible.

The research leading to these results has received funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

R. Carli is with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106, USA, carlirug@engr.ucsb.edu. S. Zampieri is with the Department of Information Engineering, Università di Padova, Via Gradenigo 6/a, 35131 Padova, Italy zampi@dei.unipd.it.

II. Problem formulation and proposed solution

Assume we have N units and that each unit i has a clock which is an oscillator able to periodically increment a register 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 register is $\tau_i(t) = \lfloor \frac{t-t_{0i}}{\Delta_i} \rfloor$, where the "floor" $\lfloor a \rfloor$ indicates the largest integer smaller than or equal to a, and t_{0i} denotes the time when the clock has been started. The unit has to use these ticks in order to estimate time. Since only the nominal clock period Δ is known, the natural time estimate is

$$y_i(t) = \Delta \tau_i(t) + y_i(t_{0i}) \tag{II.1}$$

where $y_i(t_{0i})$ is the initial offset which is an estimate of t_{0i}. Since the Δ_i's are all different, 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. Therefore some sort of information exchange and clock control must be enforced to obtain and maintain synchronization among all nodes. If we assume that the nodes exchange their clock readings $y_i(t)$ at times $t = t_h$, where $h = 0, 1, \ldots$ and $t_h < t_{h+1}$, then they can use them to adjust 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. As a preliminary step, let us observe that the evolution of $y_i(t_h)$ in (II.1) can be described through the following iterative algorithm

$$\begin{aligned} x_i(t_{h+1}) &= \begin{bmatrix} 1 & \Delta\delta_i(t_h) \\ 0 & 1 \end{bmatrix} x_i(t_h), \;\; x_i(0) = \begin{bmatrix} y_i(0) \\ 1 \end{bmatrix} \\ y_i(t_h) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(t_h) \end{aligned}$$

where $x_i(t_h) \in \mathbb{R}^2$ and where $\delta_i(t_h) := \tau_i(t_{h+1}) - \tau_i(t_h)$. 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.

Each node can use any information it receives from the neighboring nodes at time t_h to insert a control in the previous iterative algorithm

$$x_i(t_{h+1}) = \begin{bmatrix} 1 & \Delta\delta_i(t_h) \\ 0 & 1 \end{bmatrix} x_i(t_h) + u_i(t_h) \tag{II.2}$$

$$y_i(t_h) = \begin{bmatrix} 1 & 0 \end{bmatrix} x_i(t_h). \tag{II.3}$$

Notice that the previous system corresponds to the output of a second order integrator with unknown parameter, since Δ_i is not known. Moreover, the dynamics of each clock is different since in general $\Delta_i \neq \Delta_j$.

We propose here a linear control law of the following structure

$$u_i(t_h) = -F \sum_{j=1}^{N} k_{ij}(t_h) x_j(t_h) \tag{II.4}$$

where $k_{ij}(t_h)$ is the $i{-}j$ entry of the matrix $K(t_h) \in \mathbb{R}^{N\times N}$ and F is a 2×2 matrix. Notice that at time t_h the protocol requires the transmission of the state $x_j(t_h)$ from the node j to the node i if and only if $k_{ij}(t_h) \neq 0$. The problem is to determine the matrix F and the matrices $K(t_h)$, $h = 0, 1, \ldots$, such that all the $y_i(t_h)$'s converge to the same ramp shaped function.

We introduce now two simplifications which seem rather unrealistic at the moment, but which yield the starting point for a more realistic case. First assume that $t_h = hT$ where $T \in \mathbb{R}_{>0}$ is the sample period. Second assume that $K(t_h) = K$ for all $h = 0, 1, \ldots$, where K is such that $K\mathbb{1} = 0$, being $\mathbb{1}$ is N-dimensional column vector with all entries equal to 1. The condition $K\mathbb{1} = 0$ encodes the fact that, if all clocks are synchronized, then no correction is needed. Finally we will assume for simplicity that K is symmetric.

Then we obtain

$$\begin{aligned} &x_i((h+1)T) = \\ &\quad = \begin{bmatrix} 1 & \Delta\delta_i(hT) \\ 0 & 1 \end{bmatrix} \left[x_i(hT) - \sum_{j=1}^{N} K_{ij} F x_j(hT) \right]. \end{aligned}$$

Notice that $\delta_i(hT) = T/\Delta_i + \epsilon(h)$ where $-1 < \epsilon(h) < 1$ and so $\epsilon(h)$ can be neglected if $T/\Delta_i >> 1$ which will be assumed in the sequel. Moreover, in the following we denote Δ/Δ_i by d_i and we refer to d_i as the speed of the i-th clock. Introduce now the $2N$ dimensional vector $x(t_h)$ having $x'(t_h) := [x_1'(t_h), \ldots, x_N'(t_h)]^\top$ as the first N entries and having $x''(t_h) := [x_1''(t_h), \ldots, x_N''(t_h)]^\top$ as the second N entries. Then the previous equations can be collected in the following

$$\begin{aligned} &x((h+1)T) = \\ &= \begin{bmatrix} I & TD \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}K & f_{12}K \\ f_{21}K & f_{22}K \end{bmatrix} \right) x(hT) \end{aligned} \tag{II.5}$$

where I is the $N \times N$ identity matrix, f_{ij} are the entries of F and $D = \operatorname{diag}\{\Delta/\Delta_1, \ldots, \Delta/\Delta_N\} = \{d_1, \ldots, d_N\}$, where the symbol $\operatorname{diag}\{A_1, A_2, \ldots, A_N\}$ denotes the square block matrix having the square matrices A_i on the diagonal.

Our objective is to find K and F such that the synchronization error defined as

$$e(h) = [\Omega \;\; 0] x(t_h) \tag{II.6}$$

with $\Omega = I - \frac{1}{N}\mathbb{1}\mathbb{1}^*$, converges to zero while the components of $x'(h)$ follow asymptotically a ramp function, i.e., $\lim_{h\to\infty}[x'(h) - (ah + b)\mathbb{1}] = 0$ for some $a \in \mathbb{R}_{>0}$, $b \in \mathbb{R}$.

Therefore the problem we tackle in this paper can be formulated as follows. Determine K and F such that:

a) system (II.5) has one eigenvalue in 1 with algebraic multiplicity 2 and geometric multiplicity 1. This ensures that the state trajectories contains the modes of the form $ah + b$;
b) the two modes associated with the eigenvalue 1 are unobservable with respect to output e defined in (II.6);
c) all the other eigenvalues are inside the open unit disk.

Since K is assumed to be symmetric, then it can be diagonalized by an orthonormal matrix U, i.e., $U^* K U = \Lambda$, where

$$\Lambda = \operatorname{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}, \quad \lambda_i \in \mathbb{R}, \quad i = 1, \ldots, N,$$

and where $\lambda_1, \ldots, \lambda_N$ denote the eigenvalues of K. Observe that, from the assumption that $K\mathbb{1} = 0$, zero is an eigenvalue associated to the normalized eigenvector $N^{-1/2}\mathbb{1}$. Without loss of generality it is assumed that $\lambda_1 = 0$ and hence the first column of U is $N^{-1/2}\mathbb{1}$. From this it follows also that

$$U^* \left(I - \frac{1}{N}\mathbb{1}\mathbb{1}^* \right) U = \text{diag}\,\{0, I_{N-1}\}$$

where I_{N-1} denotes the $N-1$ dimensional identity matrix.

Now it is convenient to perform a change of variable so that the new state is $\bar{x}(t) := \text{diag}\,\{U^*, U^*\}x(t)$ and the new output is $\bar{e}(t) = U^* e(t)$. The new system becomes

$$\begin{aligned} &\bar{x}((h+1)T) = \\ &= \begin{bmatrix} I & TW \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\Lambda & f_{12}\Lambda \\ f_{21}\Lambda & f_{22}\Lambda \end{bmatrix} \right) \bar{x}(hT) \\ &\bar{e}(t) = \text{diag}\,\{0, I_{N-1}\}\,[I \;\; 0]\,\bar{x}(t) \end{aligned} \qquad \text{(II.7)}$$

where $W := U^* D U$.

Proposition 2.1: Consider the linear system (II.5) and (II.6). Then, for all matrices K, F and D, one is an eigenvalue of the matrix

$$\begin{bmatrix} I & TD \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}K & f_{12}K \\ f_{21}K & f_{22}K \end{bmatrix} \right) \qquad \text{(II.8)}$$

with algebraic multiplicity at least two.

Proof: The matrix in (II.8) has the same characteristic polynomial as the matrix

$$\begin{bmatrix} I & TW \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\Lambda & f_{12}\Lambda \\ f_{21}\Lambda & f_{22}\Lambda \end{bmatrix} \right). \qquad \text{(II.9)}$$

Since $\lambda_1 = 0$, it is possible to see that the matrix

$$\begin{bmatrix} I & TW \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\Lambda & f_{12}\Lambda \\ f_{21}\Lambda & f_{22}\Lambda \end{bmatrix} \right) - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

has the first column and the $N+1$-th row that are equal to zero and so its characteristic polynomial has a double roots in zero. From this the thesis follows. ■

We have seen in the previous proposition that the system (II.5) and (II.6) has one eigenvalue in 1 with algebraic multiplicity 2. In the next two subsections we provide conditions on K and F ensuring that the geometric multiplicity of the eigenvalue 1 is one and that conditions (b) and (c), stated above and related to the unobservability of the eigenvalue 1 and to the stability of system (II.5), are satisfied.

A. *Geometric multiplicity of the eigenvalue* 1 *and its unobservability*

Let us assume that K, F have been chosen so that the system (II.5) and (II.6) have one eigenvalue equal to 1 of algebraic multiplicity 2 and that all the others eigenvalues are inside the open unit circle. We will see in the next subsection how to make the latter condition satisfied. It is easy to see that the vector $v = \left[\begin{smallmatrix} \mathbb{1} \\ 0 \end{smallmatrix}\right]$ is an eigenvector associated with the eigenvalue 1. Moreover this is not observable since $[\Omega \;\; 0]v = 0$. If there exists an generalized eigenvector w such that

$$\left(\begin{bmatrix} I & TD \\ 0 & I \end{bmatrix} - \begin{bmatrix} I & TD \\ 0 & I \end{bmatrix} \begin{bmatrix} f_{11}K & f_{12}K \\ f_{21}K & f_{22}K \end{bmatrix} - \right. \left. - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \right) w = v$$

and such that $[\Omega \;\; 0]w = 0$, then we have proved both that the geometric multiplicity of the eigenvalue 1 is one and that w is not observable. Partition $w = \left[\begin{smallmatrix} w' \\ w'' \end{smallmatrix}\right]$ and notice that $[\Omega \;\; 0]w = 0$ implies that $w' = \alpha\mathbb{1}$ for some $\alpha \in \mathbb{R}$. From the previous equation we have that

$$\begin{aligned} &TDw'' + (f_{11}I + f_{21}TD)Kw' + (f_{12}I + f_{22}TD)Kw'' = \mathbb{1} \\ &f_{21}Kw' + f_{22}Kw'' = 0 \end{aligned}$$

Using the fact that $w' = \alpha\mathbb{1}$ we obtain that

$$\begin{aligned} &TDw'' + (f_{12}I + f_{22}TD)Kw'' = \mathbb{1} \\ &f_{22}Kw'' = 0 \end{aligned} \qquad \text{(II.10)}$$

Observe now that, if $f_{22} \neq 0$ and K has rank $N-1$ (which is necessary for stability as we will see in the next subsection), then $w'' = \beta\mathbb{1}$ for some $\beta \in \mathbb{R}$; in this case we have that $\beta TD\mathbb{1} = \mathbb{1}$ which is possible only if $D = (\beta T)^{-1}I$. This implies that, in case D is not of the form $D = dI$ for some $d \in \mathbb{R}$, then we have to impose that $f_{22} = 0$.

Finally in order to obtain w'' we need that the linear equation $(TD + (f_{12} + f_{22}TD)K)w'' = \mathbb{1}$ is solvable. This happens if $\det(TD + (f_{12} + f_{22}TD)K) \neq 0$. Assuming that D is a small enough perturbation of a matrix dI, we have that the last condition is equivalent to $\det(TdI + (f_{12} + f_{22}Td)K) \neq 0$ and, in turn, to $-\frac{Td}{f_{12}+f_{22}Td} \notin \sigma(K)$.

In the rest of the paper, we assume that

$$f_{12} = 0 \qquad \text{and} \qquad f_{22} = 0. \qquad \text{(II.11)}$$

Indeed, based on the above discussion, conditions in (II.11) guarantee that, if $d_i \neq 0$ for all $i \in \{1, \ldots, N\}$, then both equations in (II.10) are satisfied.

Remark 2.2: Conditions in (II.11), imply that, when nodes communicate with each other, they have to transmit only the information related to the state x'. In other words the matrix F can be regarded as a 2×1 dimensional matrix and the input $u_i(hT)$, for all $i \in \{1, \ldots, N\}$, can be rewritten as

$$\begin{aligned} u_i(hT) &= \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} \sum k_{ij} x'_j(hT) \\ &= \begin{bmatrix} f_{11} \\ f_{21} \end{bmatrix} \sum k_{ij} \left(x'_j(hT) - x'_i(hT) \right) \end{aligned} \qquad \text{(II.12)}$$

where the last equality follows from the fact that $K\mathbb{1} = 0$.

Remark 2.3: The fact that we need to impose $f_{22} = 0$ in order to ensure the unobvervability of the eigenvalue one, makes the strategy proposed by Scardovi and Sepulchre [10] for obtaining consensus for higher order systems not applicable for time synchronization. Indeed their method is strongly based on the assumption that $f_{21} = 0$ and that $f_{22} \neq 0$ which implies that, in the non ideal case where $D \neq d\mathbb{1}$, the eigenvalue 1 becomes observable and hence that consensus is not achievable.

B. *Stability*

In this subsection we consider the stability of system (II.5). We perform our analysis by assuming that D is an small perturbation of the matrix dI, for some $d \in \mathbb{R}$. Accordingly, we will design K and F only for $D = dI$. From the fact that the eigenvalues of system (II.5) depend continuously on the matrix D, it will follow that this choice of K and F will yield the stability also for a small enough perturbation of D.

Proposition 2.4: Let $D = dI$ and $f_{12} = f_{22} = 0$. Then the system linear system (II.5), besides the eigenvalue 1 with algebraic multiplicity 2, has all the remaining eigenvalues inside the open unit circle if and only if

$$\begin{aligned} & f_{11} > 0, \quad f_{21} > 0, \qquad \text{(II.13)} \\ & 0 < \lambda_i < \frac{4}{2f_{11} + Tdf_{21}}, \qquad 2 \le i \le N. \end{aligned}$$

Proof: If $D = dI$, then $W = dI$ and hence all the blocks of the matrices in (II.9) are diagonal. Therefore the system represented by (II.9) can be decoupled into N two-dimensional subsystems. It follows that the characteristic polynomial of (II.9) is given by

$$\prod_{i=1}^{N} [(z-1)^2 + \lambda_i(\beta(z-1) + \alpha)]$$

where $\beta := f_{11} + Tdf_{21}$ and $\alpha := Tdf_{21}$. Observe that for $i = 1$ we have $\lambda_i = 0$ and we get the two eigenvalues equal to 1. For the remaining eigenvalues of K we need to impose that both the roots of the polynomial $(z-1)^2 + \lambda_i(\beta(z-1) + \alpha)$ has to be inside the open unit circle. Using the bilinear transformation this is equivalent to have that

$$(\lambda(\alpha - 2\beta) + 4)s^2 + 2\lambda(\beta - \alpha)s + \alpha\lambda$$

is Hurwitz-stable (i.e., it has both roots with negative real part). This happens if and only if the coefficients have the same sign. It can be seen that the coefficients can not be all negative and that they are all positive if and only if $0 < \alpha < \beta$ and $0 < \lambda_i < \frac{4}{2\beta - \alpha}$, $i = 2, .., N$. These inequalities are equivalent to (II.13). ■

Remark 2.5: Assume that we are given a graph $G = (V, E)$ where $V = \{1, \ldots, N\}$ and where $E \subseteq V \times V$. Assume that G is undirected, namely that $(i,j) \in E$ if and only if $(j,i) \in E$. This graph describes the feasible communications, i.e., the node j can transmit information to the node i if and only if $(j,i) \in E$. For this reason we say that a matrix K is compatible with G if $k_{ij} \neq 0$ implies $(j,i) \in E$. Assume now that G is connected. Then values of f_{11}, f_{21} and a matrix K compatible with G yielding clock synchronization can be selected in a completely distributed way. To see this let us consider the symmetric stochastic matrix P built according to the Metropolis weights technique [13] as follows

$$P_{ij} = \begin{cases} \frac{1}{\max\{d_i, d_j\}} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j \\ 1 - \sum_{j \neq i} P_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \qquad \text{(II.14)}$$

where d_i denotes the number of neighbors of the node i, i.e., $d_i = |\mathcal{N}_i|$ and $\mathcal{N}_i = \{j \in V | (j,i) \in \mathcal{E}, i \neq j\}$. Note that P is a stochastic symmetric matrix. Moreover it is also easy to see that P is primitive and hence, by the Perron-Frobenius theorem, one eigenvalue of P is 1 and all the others are inside the open interval $]-1, 1[$. Then the matrix $K := I - P$, and the parameters $f_{11} := 1/2$ and $f_{21} := 1/(2Td)$ satisfy the conditions of Proposition 2.4. Other distributed strategies are possible for designing the matrix K such as the Laplacian matrix [8].

III. Robustness analysis

In the previous section we have proposed a method for determining K and F based on the continuity of the eigenvalues as functions of the parameters. In this section we perform a more refined analysis based on H^∞ control techniques. Again we consider the case where $f_{12} = f_{22} = 0$.

We start our analysis by assuming that $D = dI + \overline{\eta}\bar{\Delta}$ where $\bar{\Delta}$ is a diagonal matrix with entries belonging to $[-1, 1]$. Then, $\overline{\eta}$ describes the amplitude of the allowed perturbation of dI. Moreover we assume that also the knowledge of K is uncertain, namely that K is any symmetric matrix of rank $N-1$ such that $K\mathbb{1} = 0$ and such that all the nonzero eigenvalues belong to the interval $[\lambda_m, \lambda_M]$, where $0 < \lambda_m \le \lambda_M$. The objective is to use the small gain theorem in order to understand the stability of the previous system. Observe that we are not interested into the asymptotic stability of system (II.5) and (II.6), but into something weaker. However, in order to apply the standard small gain theorem, we need to transform our problem into a standard stability problem, and, to this aim, we need to make a suitable change of variable.

Consider any unitary matrix U having $N^{-1/2}\mathbb{1}$ as first column. Then

$$\bar{K} := U^* K U = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{K} \end{bmatrix}$$

where all the eigenvalues of $\bar{K}$ belong to the interval $[\lambda_m, \lambda_M]$. Let us introduce now the variable $\bar{x}(t) := \text{diag}\{U^*, U^*\}x(t)$ obtaining the following system

$$\begin{aligned} & \bar{x}(h+1) = \\ & \quad = \begin{bmatrix} I & TW \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\bar{K} & 0 \\ f_{21}\bar{K} & 0 \end{bmatrix} \right) \bar{x}(h) \end{aligned} \qquad \text{(III.1)}$$

where $W := U^* D U$. Observe that here, with a slight abuse, we use the same notation used in (II.7), even though (II.7) could be, in general, different from (III.1), since, in this section, the diagonalization of K is not needed.

As noticed in Proposition 2.1 the matrix

$$\begin{bmatrix} I & TW \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\bar{K} & 0 \\ f_{21}\bar{K} & 0 \end{bmatrix} \right) - \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

has the first column and the $N+1$-th row that are equal to zero and so its characteristic polynomial has a double roots in zero. From this it follows that two eigenvalues of system (III.1) are equal to 1 and the remaining $2N-2$ eigenvalues coincide with the eigenvalues of the submatrix obtained by deleting the 1^{st} and the $(N+1)^{\text{th}}$ rows and columns of the matrix appearing in (III.1). It can be seen that this submatrix is given by

$$\begin{bmatrix} I & \tilde{T}W \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\tilde{K} & 0 \\ f_{21}\tilde{K} & 0 \end{bmatrix} \right)$$

where here I means the $(N-1) \times (N-1)$ identity matrix. Now we want to study the stability of the stability of the following system

$$\begin{aligned} & \tilde{x}(h+1) = \qquad \text{(III.2)} \\ & \quad = \begin{bmatrix} I & \tilde{T}W \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} f_{11}\tilde{K} & 0 \\ f_{21}\tilde{K} & 0 \end{bmatrix} \right) \tilde{x}(h) \end{aligned}$$

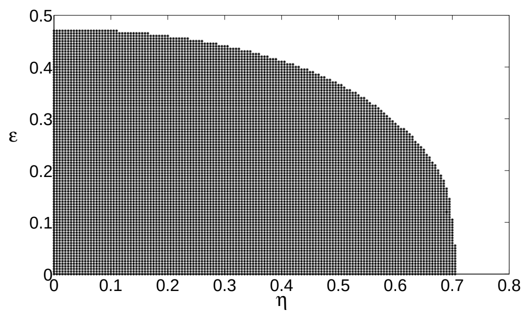

Fig. 1. In the Figure it is represented the set of pairs $(\bar{\eta}, \bar{\epsilon})$ ensuring that system (III.3) and (III.4) is asymptotically stable.

where $\tilde{x}(h)$ is a $N-2$ dimensional state. To this aim we will use the small gain theorem.

Let $\bar{\lambda} := \frac{\lambda_m+\lambda_M}{2}$ and $\bar{\epsilon} := \frac{\lambda_M-\lambda_m}{\lambda_M+\lambda_m}$ and $\tilde{H} := \frac{1}{\bar{\epsilon}}\left(\frac{1}{\bar{\lambda}}\tilde{K}-I\right)$. Let moreover $\tilde{V} := \frac{1}{\bar{\eta}}(\tilde{W}-dI)$. Notice that $\tilde{H}$ is a symmetric matrix with eigenvalues belonging to $[-1,1]$ and so $||\tilde{H}||_2 \leq 1$. Notice moreover that $\tilde{V}$ is a submatrix of $U^*\Delta U$ and so $||\tilde{V}||_2 \leq 1$. We want to quantify how big $\bar{\epsilon}$ and $\bar{\eta}$ are, while maintaining the system stability. It can be shown that the system

$$\begin{aligned}\tilde{x}(h+1) &= \begin{bmatrix} I & TdI \\ 0 & I \end{bmatrix}\tilde{x}(h) + \begin{bmatrix} I & TdI \\ 0 & I \end{bmatrix}\tilde{u}(h) + \\ &\qquad + \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix}\tilde{v}(h) \\ \tilde{y}(h) &= \begin{bmatrix} I & 0 \end{bmatrix}\tilde{x}(h) + \begin{bmatrix} I & 0 \end{bmatrix}\tilde{v}(h) \\ \tilde{z}(h) &= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}\tilde{x}(h) + \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}\tilde{u}(h) \end{aligned} \qquad \text{(III.3)}$$

with the feedback laws

$$\begin{aligned}\tilde{v}(h) &= \begin{bmatrix} \bar{\epsilon}I & 0 \\ 0 & \bar{\eta}I \end{bmatrix}\begin{bmatrix} \tilde{H} & 0 \\ 0 & \tilde{V} \end{bmatrix}\tilde{z}(h) \\ \tilde{u}(h) &= \begin{bmatrix} -f_{11}\bar{\lambda}I \\ -f_{21}\bar{\lambda}I \end{bmatrix}\tilde{y}(h) \end{aligned} \qquad \text{(III.4)}$$

is equivalent to (III.2). Notice that the first feedback models the system uncertainty, while the second feedback contains the design parameters. From the small gain Theorem, we have that, if the product between the $\mathcal{L}_2$ gain of the linear system (III.3) (i.e., the system having $\tilde{x}$ as state, $\left[\begin{smallmatrix}\tilde{y}\\ \tilde{z}\end{smallmatrix}\right]$ as output, $\left[\begin{smallmatrix}\tilde{u}\\ \tilde{v}\end{smallmatrix}\right]$ as input) and the $\mathcal{L}_2$ gain of the map (III.4) (i.e., the map that gives $\left[\begin{smallmatrix}\tilde{u}\\ \tilde{v}\end{smallmatrix}\right]$ as function of $\left[\begin{smallmatrix}\tilde{y}\\ \tilde{z}\end{smallmatrix}\right]$) is strictly smaller than one, then system (III.2) is asymptotically stable. We can use this fact to provide numerically a set of pairs $(\bar{\eta}, \bar{\epsilon})$ yielding the stability of (III.2).

Example 3.1: In this example we assume that $\bar{\lambda} = 1$, $d = 1$, $T = 100$, $f_{11} = 1/2$, $f_{21} = 1/(2dT) = 1/200$. In Figure 1 we plot in black the region of pairs $(\bar{\eta}, \bar{\epsilon})$ for which the product of the two aforementioned $\mathcal{L}_2$ gains is strictly smaller than 1.

IV. Pseudo-synchronous implementation

In the previous section we have seen that the control law described in (II.12) asymptotically synchronizes system (II.5), provided that the matrix D is a small perturbation of a matrix dI, $d \in \mathbb{R}_{>0}$, and that conditions (II.13) are satisfied. The goal of this section is to analyze a more realistic implementation of the proposed synchronization method.

We start by observing that, in any possible implementation of a synchronization algorithm, there are two important limitations that have to be taken into account. The first limitation is related to the existence of delays between the time a message is prepared by the transmitting node in order to be sent, and the time in which this message is used by the receiving node in its synchronization updating step. The second limitation concerns the fact that the nodes data transmissions and algorithm updating steps cannot occur exactly at the same time, i.e., synchronously, as described in (II.12). 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 this Section we assume that the delays are negligible and we concentrate only on the second limitation we discussed above. We will turn our attention to communication delays in Subsection IV-A and in Section V.

Next, we propose a version of the synchronization algorithm which allows the nodes to overcome the fact that, in a realistic setup, they are unable to carry out their transmission and updating actions synchronously. This algorithm is a suitable modification of (II.5) and (II.12) and, in what follows, we refers to it as the *pseudo-synchronous* implementation of (II.5) and (II.12). First we specify how the nodes select the transmission and the updating time instants. To do so, we need the following notations.

- By N_i, we denote the set of neighbors of the node i, i.e., $N_i = \{j \,|\, k_{ij} \neq 0, j \neq i\}$, or, equivalently, since K is symmetric, $N_i = \{j \,|\, k_{ji} \neq 0, j \neq i\}$.
- By $t_{tx,k,i}$ we denote the time instant in which the i-th node performs its k-th transmission of information; here we assume a broadcast model, i.e., each node transmits, at the same time, the same information to all its neighbors;
- By $t_{up,k,i}$ we denote the time instant in which the i-th node performs its k-th updating step.

Moreover, given the time t, with the symbol t^+ we will mean the time just after t. Clearly, $t_{tx,k,i}$ ad $t_{up,k,i}$ can be determined by the node i relying only on its local information. Based on this observation we define:

- $t_{tx,k,i}$ as the first time such that $x'_i(t_{tx,k,i}) = kT$, namely the node i transmits when its time estimate is equal to kT for the first time;
- $t_{up,k,i}$ to be equal to
$$\max\{t_{tx,k,h} | h \in N_i \cup \{i\}\},$$
namely the node i updates its state only after the completion of all the communication actions it is involved, included the transmission.

The *pseudo-synchronous* algorithm can be formally described as follows.

Processor states: For each $i \in \{1,\ldots,N\}$, the node i stores in its memory the state variables x'_i, x''_i, the parameters f_{11}, f_{12} and the weights k_{ij}, $j \in N_i$.

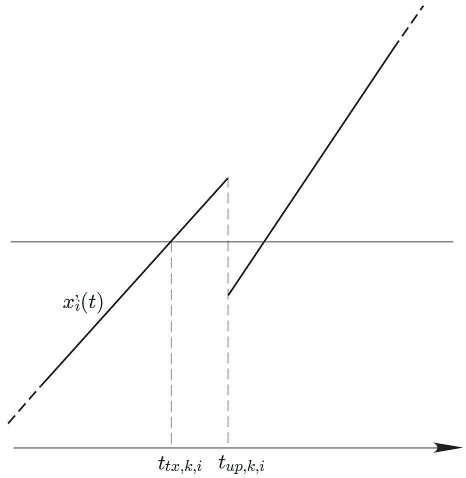

Fig. 2. In this Figure we show how the time instants $t_{tx,k,i}$ and $t_{up,k,i}$ are related to the state evolution $x'_i(t)$.

Initialization: For $i \in \{1,\dots,N\}$, the variables $x'_i(0)$ are initialized to $y_i(0)$, while the variables $x''_i(0)$ are initialized to 1.

Transmitting and updating steps: During the k-th iteration of the algorithm each node i, $i \in \{1,\dots,N\}$ performs the following actions:

1. for $t \in]t_{up,k-1,i}, t_{up,k,i}[$, the node i computes its state according to

$$x_i(t) = \begin{bmatrix} 1 & \Delta(\tau_i(t) - \tau_i(t_{up,k-1,i})) \\ 0 & 1 \end{bmatrix} x_i(t^+_{up,k-1,i}).$$

Similarly to Section II, we approximate $\Delta(\tau_i(t) - \tau_i(t_{up,k-1,i}))$ by $d_i(t - t_{up,k-1,i})$.

2. At instant time $t_{tx,k,i}$ the node i broadcasts to its neighbors the state $x'_i(t_{tx,k,i})$ which concides with kT.

3. For all $j \in N_i$, at time $t_{tx,k,j}$ the node i receives from the node j the value $x'_j(t_{tx,k,j}) = kT$; it computes the difference $x'_j(t_{tx,k,j}) - x'_i(t_{tx,k,j}) = kT - x'_i(t_{tx,k,j})$ and, it stores it in memory.

4. At time $t_{up,k,i}$ the node i computes the input $u_i(t_{up,k,i})$ using all the stored differences $x'_j(t_{tx,k,j}) - x'_i(t_{tx,k,j})$, $j \in N_i$, namely

$$u_i(t_{up,k,i}) = \begin{bmatrix} f_{11} \\ f_{12} \end{bmatrix} \sum_{j \in N_i} k_{ij} \left(x'_j(t_{tx,k,j}) - x'_i(t_{tx,k,j}) \right). \tag{IV.1}$$

Accordingly it updates its state x_i as

$$x_i(t^+_{up,k,i}) = x_i(t_{up,k,i}) + u_i(t_{up,k,i})$$

It must be noticed that $u_i(t_{up,k,i})$ could be such that $x_i(t^+_{up,k,i}) > (k+1)T$. In this case there would not be a time $t_{tx,k+1,i}$ such that $x_i(t_{tx,k+1,i}) = (k+1)T$. If $x_i(t^+_{up,k,i}) > (k+1)T$ occurs we assume that the i-th node performs its $k+1$-th transmission just after its k-th updating step, sending to all its neighbors the estimate $x_i(t^+_{up,k,i})$.

Figure 2 illustrate the relation between the graph of the time estimation $x'_i(t)$ and the time instants $t_{tx,k,i}$ and $t_{up,k,i}$.

We show now that the *pseudo-synchronus algorithm* can be written as a perturbation of the synchronous algorithm (II.5). To do so, we first need to define the sampling instants since the pseudo-synchronous algorithm is continuous time while the synchronous algorithm is discrete time. A suitable definition is given by

$$t_k = \min_i \{t_{tx,k,i}\}.$$

Moreover let

$$\begin{aligned} \delta T_k &= t_{k+1} - t_k - T \\ \delta t_{up,k} &= [t_{up,k,1} - t_k, \dots, t_{up,k,N} - t_k]^T \\ \delta t_{tx,k} &= [t_{tx,k,1} - t_k, \dots, t_{tx,k,N} - t_k]^T \end{aligned}$$

Loosely speaking t_k represents the first time instant in which the estimate of a node reaches the value kT, $\delta t_{up,k}$ and $\delta t_{tx,k}$ accounts for the lack of synchronicity in performing the updating and transmitting actions, $t_{k+1} - t_k - T$ measures the length of the k-th sample period with the respect to the nominal length T. The following proposition shows how the pseudo-synchronus algorithm can be written as a discrete time nonlinear system that is a perturbation of linear system representing the synchronous algorithm. In the statement of the theorem we use the notation $\mathrm{diag}\,\{v\}$, where v is a vector, to mean the diagonal matrix having the components of v on the diagonal.

Proposition 4.1: Consider the *pseudo-synchronous* algorithm illustrated above. Then

$$x(t_{k+1}) = Ax(t_k) + \Phi(x(t_k)) \tag{IV.2}$$

where the matrix A describes the linear iteration relative to the synchronous implementation and where

$$\Phi(x(t_k)) = \begin{bmatrix} \Phi'(x'(t_k), x''(t_k)) \\ \Phi''(x'(t_k), x''(t_k)) \end{bmatrix} \tag{IV.3}$$

is defined as

$$\begin{aligned} \Phi'(x'(t_k), x''(t_k)) &:= \mathrm{diag}\,\{\delta t_{up,k}\}\, D\, (x''(t_k) - x''(t_{k+1})) \\ &\quad + f_{11}\,(\mathrm{diag}\,\{K\,\delta t_{tx,k}\} - K\mathrm{diag}\,\{\delta t_{tx,k}\})\, Dx''(t_k) \\ &\quad + \delta T_k D x''(t_{k+1}) \end{aligned}$$

and

$$\begin{aligned} &\Phi'(x'(t_k), x''(t_k)) := \\ &\qquad f_{21}\,(\mathrm{diag}\,\{K\,\delta t_{tx,k}\} - K\mathrm{diag}\,\{\delta t_{tx,k}\})\, Dx''(t_k). \end{aligned}$$

Observe that the above system can be written as the sum of the synchronous system (II.5) and some perturbative terms which arise since the clocks do not carry out synchronously their transmitting and updating actions. The following result characterizes the asymptotic synchronization of the system (IV.2).

Theorem 4.2: Let $\mathbb{1}$ and $\mathbf{0}$ be the N-dimensional vectors with all the components equal to 1 and 0, respectively. Consider the system (IV.2) and let $e(t_k)$ be the synchronization error. Then for any $\epsilon > 0$, there exist a neighborhood $U \in \mathbb{R}^N$ of $\mathbb{1}$ and a neighborhood $W \subseteq \mathbb{R}^N$ of $\mathbf{0}$ such that, if $x'(0) \in W$ and $[d_1, \dots, d_N]^T \in U$, then the following three properties hold true

a) $x'(t^+_{up,k,i}) < (k+1)T$ for all $k \geq 1$ and for all $i \in \{1, \ldots, N\}$;
b) $\lim_{k\to\infty}(t_{k+1} - t_k) = \bar{T}$ where $|\bar{T} - T| \leq \epsilon$;
c) the synchronization error converges exponentially fast to 0, i.e., there exists $C > 0$ and $0 \leq \rho < 1$ such that

$$||e(t_k)|| \leq C\rho^k ||e(0)||.$$

The proofs of the results present in this Section are omitted for reasons of space. We only remark that the proof of Theorem 4.2 is quite involved and is obtained thanks to arguments related to the center manifold theorem. However we refer the interested reader to the document in [14].

A. Communication delays

In this subsection we model the sources of disturbances for the *pseudo-synchronous* algorithm above described. In particular we focus on the transmission delays. Let $t_{tx,k,j}$ be defined as above,i.e., the time instant in which the j-th node carries out its k-th transmission. Let $i \in N_j$ and let $t_{rx,k,i,j}$ be time instant in which the i-th clock receives the information sent by the j-th clock. A suitable model to describe $t_{rx,k,i,j}$ is given by

$$t_{rx,k,i,j} = t_{tx,k,j} + \gamma_{k,i,j}$$

where $\gamma_{k,i,j}$ is a nonnegative random variable denoting the deliver delay between j and i, of mean $\bar{\gamma}$, variance σ_γ and bounded in size by γ_M. Since the i-th node receives the information sent by the j-th node at time $t_{tx,k,j}$ at the delayed time $t_{rx,k,i,j}$ we have that the difference $x'_j(t_{tx,k,j}) - x'_i(t_{tx,k,j})$ used in (IV.1) must be replaced by $x'_j(t_{tx,k,j}) - x'_i(t_{rx,k,i,j})$. Accordingly, in this new model, Equation (IV.1) becomes

$$u_i(t_{up,k,i}) = \begin{bmatrix} f_{11} \\ f_{12} \end{bmatrix} \sum_{j\in N_i} k_{ij}\left(x'_j(t_{tx,k,j}) - x'_i(t_{rx,k,i,j})\right). \tag{IV.4}$$

If we assume that each clock knows the mean of the delays, namely the value of $\bar{\gamma}$, we could modify the above equation adding to the term $x'_j(t_{tx,k,j}) - x'_i(t_{rx,k,i,j})$ a correcting term trying to compensate the effects due to the delivery delay. More precisely, observe that

$$\begin{aligned} x'_i(t_{tx,k,j}) &= x'_i(t_{rx,k,i,j} - \gamma_{k,i,j}) \\ &= x'_i(t_{rx,k,i,j}) - \gamma_{k,i,j} d_i x''_i(t_{rx,k,i,j}), \end{aligned}$$

where the unknown quantities $\gamma_{k,i,j}$ and d_i can be suitably approximated by $\gamma_{k,i,j} \approx \bar{\gamma}$ and $d_i \approx 1$[1]. In this way we obtain

$$x'_i(t_{tx,k,j}) \approx x'_i(t_{rx,k,i,j}) - \bar{\gamma} x''_i(t_{rx,k,i,j})$$

and, thereby,

$$\begin{aligned} u_i(t_{up,k,i}) = \begin{bmatrix} f_{11} \\ f_{12} \end{bmatrix} \sum_{j\in N_i} k_{ij}\big(x'_j(t_{tx,k,j}) - x'_i(t_{rx,k,i,j}) + \\ + \bar{\gamma} d_i x''_i(t_{rx,k,i,j})\big). \end{aligned} \tag{IV.5}$$

It is worth remarking that, besides the transmission delays, there are two additional sources of disturbance. The first is due to the fact the local clock has a limited resolution. Indeed, in the analysis we previously performed, given two time instants t_1 and t_2, $t_1 < t_2$, and the clock i, we approximated the quantity $\Delta(\tau_i(t_2) - \tau_i(t_1))$ by $d_i(t_2 - t_1)$. Observe that $d_i(t_2 - t_1)$ is a continuous quantity of the time difference $t_2 - t_1$, while $\Delta(\tau_i(t_2) - \tau_i(t_1))$ is a quantized quantity belonging to the alphabet $\{0, \Delta, 2\Delta, \ldots\}$. As a consequence, the quantization can be approximated as a fictitious measurement noise v_i that enters the estimation dynamics as $\Delta(\tau_i(t_2) - \tau_i(t_1)) = d_i(t_2 - t_1) + v_i(t_2)$. It is common practice to approximate quantization error as white noise of zero mean and variance $r = \Delta^2/12$. The second source of disturbance is that the local clock speed d_i is not constant but can change over time due to temperature changes or other effects. A better model for the clock speed is given by $d_i(t_{k+1}) = d_i(t_k) + n_i(t_k)$ where $n_i(t_k)$ is a zero mean white noise with a certain variance q.

In this paper we do not provide a theoretically analysis of the effects of the aforementioned disturbances on the performance of our *pseudo-synchronous* algorithm. We limit ourselves to provide a numerical example related to the transmission delays in the next Section. However, we remark that, thanks to the linearity of our synchronization strategy, transmission delays, quantization, and time-varying clock speeds, can be incorporated into the model of our *pseudo-synchronous* algorithm as additive noises. This fact should make their analysis tractable. By contrast the protocol in [6], which is based on the cascade of two distributed least-squares algorithms, and the protocol in [7], which is based on the cascade of two first order consensus algorithms, are both nonlinear and do not lead to simple characterization of performance in presence of noise.

V. Numerical Examples

In this section we provide two examples illustrating the approach proposed in this paper. Specifically, in Example 5.1 we simulate the *pseudo-synchronous* algorithm starting from initial condition of different size, in Example 5.2 we simulate the effects of the communication delays.

Example 5.1: In this simulation we consider a connected random geometric graph generated by choosing $N = 50$ points uniformly distributed in the unit square, and then placing an edge between each pair of points at distance less than 0.4. We simulate the behavior of the *pseudo-synchronization* algorithm illustrated in Section IV. We assume that $T = 100$, $f_{11} = 1/2$, $f_{21} = 1/(2T)$ and that the matrix K is built according to the Metropolis method illustrated in Remark 2.5. The results obtained are reported in Figure 3 where we plot the trajectories of the quantity $N^{-1/2} \log \|e(t_k)\|$, generated starting from initial conditions of different size. Precisely,

- the blue line refers to initial conditions $x'_i(0)$, d_i randomly chosen inside $[0, 10]$ and $[1 - 10^{-1}, 1 + 10^{-1}]$, respectively;
- the black line refers to initial conditions $x'_i(0)$, d_i randomly chosen inside $[0, 1]$ and $[1 - 10^{-2}, 1 + 10^{-2}]$, respectively;
- the red line refers to initial conditions $x'_i(0)$, d_i randomly chosen inside $[0, 10^{-1}]$ and $[1 - 10^{-3}, 1 + 10^{-3}]$, respectively.

[1]We recall that the i-th clock knows the nominal value Δ but not the value of Δ_i and hence also of d_i. The approximation $d_i \approx 1$ is obtained by assuming that $\Delta_i \approx \Delta$.

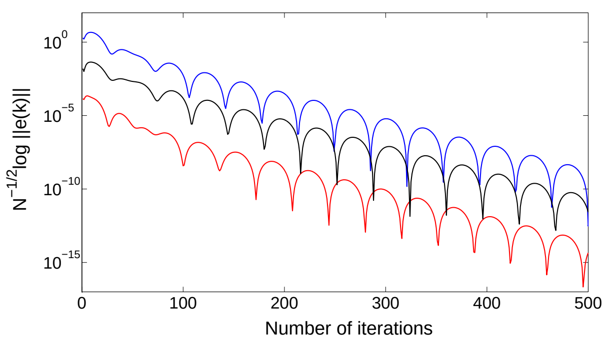

Fig. 3. Trajectories of the synchronization error generated by the *pseudo-synchronous* algorithm starting from initial conditions of different size.

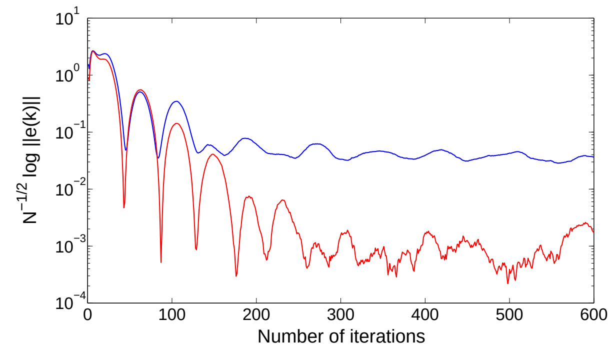

Fig. 4. Effects of the communication delays on the performance of the *pseudo-synchronous* algorithm.

The plots reported are the result of the average over 1000 Monte Carlo runs, randomized with respect to both the graph and the initial conditions. Observe that all the trajectories converge to zero with the same rate of convergence and that, as expected, the greater the initial condition is, the greater the value of $\|e(t_k)\|$ is. Moreover, we evaluated also the value of the sample period at the steady-state, i.e, the value of $\bar{T} = \lim_{k\to\infty}(t_{k+1} - t_k)$. We obtain $\bar{T} = 100.153$ for the trajectory depicted in blue, $\bar{T} = 100.042$ for the trajectory depicted in black and $\bar{T} = 100.001$ for the trajectory depicted in red. Observe that the value of $\bar{T}$ increases with the uncertainty on the initial condition.

Example 5.2: In this example we simulate the effects of the communication delays. We assume that the graph G and the matrix K are generated as in the previous Example. Moreover, again $T = 100$, $f_{11} = 1/2$, $f_{21} = 1/(2T)$. We implement both (IV.4) (blue line) and (IV.5) (red line) and the results obtained are reported in Figure 4 where we depicted the behavior of the quantity $N^{-1/2}\log\|e(t_k)\|$. The initial conditions $x'_i(0)$, d_i of the trajectories plotted have been randomly chosen inside $[0, 10]$ and $[1 - 10^{-1}, 1 + 10^{-1}]$, respectively. Moreover 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.

We assume that $\gamma_{i,j,k}$, for all $(i, j) \in G$ and for all $k > 0$, is uniformly chosen within the interval $[0, 1]$. Clearly the presence of the delays prevents our algorithm to reach the asymptotic synchronization. However observe that, at the steady-state the variable $N^{-1/2}\log\|e(t_k)\|$ oscillates within an interval whose amplitude is smaller that 10^{-1}, i.e., one order of magnitude less than the maximum value γ_M that the delays can assume. Finally we can see that the correcting term introduced in (IV.5) provided a significant improvement of the attainable performance.

VI. Conclusions

In this paper we have proposed a distributed clock synchronization algorithm based on the consensus of higher order linear systems. We have proved the stability of this algorithm in the unrealistic synchronous implementation and in a realistic pseudo-synchronous implementation. Simulations supports our belief that also the completely asynchronous implementations converges. The formal proof of this fact seems to be a quite challenging and difficult question that we hope to answer in the future.

References

[1] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timingsync protocol for sensor networks," in *Proceedings of the first international conference on Embedded networked sensor systems (SenSys'03)*, 2003.

[2] M. Maròti, B. Kusy, 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.

[3] 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.

[4] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on wireless sensor networks with coupled discrete-time oscillators*, 2007.

[5] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, November 2005.

[6] 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.

[7] L. Schenato and F. Fiorentin, "Average timesync: A consensus-based protocol for time synchronization in wireless sensor networks," in *Proceedings of 1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys09)*, September 2009.

[8] R. O. Saber, J. Fax, and R. Murray, "Consensus and cooperation in multi-agent networked systems," *Proceedings of IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[9] C.-Y. Kao, U. T. Jonsson, and H. Fujioka, "Characterization of robust stability of a class of interconnected systems," *To appear in Automatica*.

[10] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," *Automatica*, vol. 45, no. 11, pp. 2557–2562, 2009.

[11] P. Wieland, J.-S. Kim, H. Scheu, and F. Allgower, "On consensus in multi-agent systems with linear high-order agents," in *Proc. 17th IFAC World Congress*, 2008, pp. 1541–1546.

[12] W. Ren, "On consensus algorithms for double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1503–1509, 2008.

[13] L. Xiao, S. Boyd, and S. Lall, "A scheme for asynchronous distributed sensor fusion based on average consensus," in *International Conference on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, 2005, pp. 63–70.

[14] R. Carli and S. Zampieri, "Networked clock synchronization based on second order linear consensus algorithms," *Available at http://motion.mee.ucsb.edu/∼carlirug/Papers/Synchro.pdf.*