Distributed synchronization of noisy non-identical double integrators

Ruggero Carli, Alessandro Chiuso, *Senior Member, IEEE*, Luca Schenato *Member, IEEE*, Sandro Zampieri

***Abstract*— In this paper we present a novel synchronization protocol to synchronize a network of controlled discrete-time double integrators which are non-identical, with unknown model parameters and subject to additive measurement and process noise. This framework is motivated by the typical problem of synchronizing a network of clocks whose speeds are non-identical and are subject to variations. This synchronization protocol is formally studied in its synchronous implementation. In particular, we provide a completely distributed strategy that guarantees convergence for any undirected connected communication graph, and we also propose an optimal convex design strategy when the underlaying communication graph is known. Moreover, this protocol can be readily used to study the effect of noise and external disturbances on the steady-state performance. Finally, some simulations including also asynchronous implementation of the proposed algorithm are presented.**

I. Introduction

The extraordinary success of Internet and of wireless technologies has created the opportunity to interconnect a huge number of devices which can exchange information and cooperate to accomplish new tasks and to control the environment more effectively. One important problem to be solved in many applications involving a network of distributed devices, such as in wireless sensors networks (WSNs), is to maintain them temporally synchronized. The main challenges in time synchronization for networked clocks are due to random communication delay among devices, to the unknown, non-identical and often time-varying periods of each clock oscillator, and to the unknown topology of the network. Moreover, there might not be a reference clock in the network.

Many synchronization protocols have already been proposed and experimentally tested, in particular for WSNs. A common approach is to create a hierarchical structure such a directed tree where each clock synchronizes itself with respect to its parent. The challenge with this approach is to dynamically elect a root and reconstruct the tree whenever nodes fail or new nodes appear, as proposed by Ganeriwal et al. [4] and Maroti et al. [6] for example. Another hierarchical approach is to divide the network 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, as described by Elson et al. [2]. These hierarchical approaches however require substantial overhead to build the dynamic trees or clusters and might not scale well for very large networks. More recently, totally distributed synchronization protocols have been proposed where each node in the network runs exactly the same algorithm irrespective of the size or topology of the network, such as those proposed by Solis et al. [11] and Schenato et al. [10].

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. This class of algorithms has been successfully applied to many applications such as flocking, coordinated formation control and distributed estimation (e.g. see [8]). As in [11] and [10] our proposed protocol is fully distributed, however it has the advantage to use a simple linear output feedback strategy which allows performance analysis also in the presence of measurement and process noise. In fact, the protocol in [11], which is based on the cascade of two distributed least-square algorithms, and the protocol in [10], 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 the presence of 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 in 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. Synchronization can be modeled as the tracking of the average of external signals with linear growth, namely local clocks without compensation. Work in the context of signal tracking with consensus algorithms have been proposed, among others, by Spanos et al. [12], Freeman et al. [3], and Zhu et. al [16]. However these works are in continuous time with no noise and do not provide optimization strategies for the protocol parameters. The presence of noise has been explicitly taken into account by Xiao et al. [15], but only for first order consensus dynamics. Differently, synchronization of networked higher order systems is a more recent topic of research. Most of the available results are for synchronization of either non-identical systems which are strictly stable [5], or identical linear systems [9],[13]. More challenging is the problem of stabilizing unstable systems. Specific attention has also been given to the synchronization of double integrators which are unstable systems with a ramp mode [7]. However these strategies strongly rely on the assumption that all systems are identical and no optimization is performed.

In this paper we first propose a distributed clock synchronization protocol based on the consensus algorithm for non identical double integrators whose rates of growth are not known nor measurable. This technique is only analyzed in the unrealistic synchronous implementation. Motivated by this application, the main contributions of the present paper reside in the extension of the optimization techniques proposed by Xiao et al. [15] for first order consensus protocols to consensus protocols for double integrators. More specifically, we propose centralized optimization algorithms for optimally designing the protocol parameters both in terms of rate of convergence and in terms of steady state error in case the protocol is perturbed by an additive noise. Interestingly, this last optimization problem shows that the optimal design strategy requires the optimization of a convex function of all the consensus matrix eigenvalues, thus being quite different from the standard procedure which suggests to minimize the second largest eigenvalue of the consensus matrix. Finally, we include some simulations where we compare the performance of the synchronous implementation with a randomized asynchronous implementation. These simulations seem to suggest that the solution we propose is effective also in its more realistic asynchronous version, even though we do not have at the moment any theoretical evidence of this fact.

This work has been supported by the project FeedNetBack funded by the European Community, by WISE-WAI project funded by CaRiPaRo Foundation, by the national project *Sviluppo di nuovi metodi e algoritmi per l'identificazione, la stima bayesiana e il controllo adattativo e distribuito* funded by MIUR and Progetto di Ateneo CPDA090135/09 funded by the University of Padova.

Ruggero Carli is with the Center for Control, Dynamical Systems and Computation, University of California at Santa Barbara, Santa Barbara, CA 93106, USA. carlirug@engineering.ucsb.edu

Alessandro Chiuso is with the Department of Management and Engineering, Università di Padova, Stradella S. Nicola, 3 - 36100 Vicenza, Italy, ph. +39-049-8277709, fax. +39-049-8277699, e-mail: chiuso@dei.unipd.it

Luca Schenato and Sandro Zampieri are with the Department of Information Engineering, Università di Padova, Via Gradenigo, 6/b, 35100 Padova, Italy. e-mail:{schenato,zampi}@dei.unipd.it

II. MODELING AND MOTIVATIONS

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 periodically exchange their clock readings $y_i(t)$ at times $t = hT$, where $h = 0, 1, \ldots$ and $T \in \mathbb{R}$ is the synchronization period, 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 $y_i(hT)$ in (II.1) can be described through the following iterative algorithm

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

where $x_i(hT) \in \mathbb{R}^2$ and where $\delta_i(h) := \tau_i((h+1)T) - \tau_i(hT)$. 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. For sake of conciseness, with no loss of generality we will assume in the sequel that $T = 1$.

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

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

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

Notice that $\delta_i(h) = \tau_i((h+1)T) - \tau_i(hT) = 1/\Delta_i + \epsilon(h)$ where $-1 < \epsilon(h) < 1$ and so $\epsilon(h)$ can be neglected if $1/\Delta_i >> 1$ which will be assumed in the sequel. Notice moreover 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

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

where $k_{ij}(h)$ is the $i-j$ entry of the matrix $K(h) \in \mathbb{R}^{N\times N}$, $F \in \mathbb{R}^{2\times 1}$, and we assume that communication and computational delays are negligible. 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 $k_{ij}(h) \neq 0$. The problem is to determine the matrix F and the matrices $K(h)$ such that all the $y_i(h)$'s converge to the same ramp shaped function.

Assume now that $K(h) = K$ for all h and that $K\mathbb{1} = 0$, where $\mathbb{1}$ is the N-dimensional column vector with all entries equal to 1. Without loss of generality we can assume $F = [f_1 \; f_2]^T = [1 \; \alpha]^T$ since f_1 can be absorbed into K so that $\alpha := f_2/f_1$. Introduce finally the $2N$ dimensional vector $x(h)$ having $x'(h) := [x_1'(h), \ldots, x_N'(h)]^\top$ as the first N entries and having $x''(h) := [x_1''(h), \ldots, x_N''(h)]^\top$ as the second N entries. Then the previous equations can be collected in the following

$$\begin{aligned} x(h+1) &= A_D x(h), \;\; x(0) = \begin{bmatrix} x'(0) \\ \mathbb{1} \end{bmatrix} \\ A_D &:= \begin{bmatrix} I-K & D \\ -\alpha K & I \end{bmatrix} \end{aligned} \tag{II.5}$$

where $I \in \mathbb{R}^{N\times N}$ is the identity matrix, $D = \text{diag}\{\frac{\Delta}{\Delta_1}, \ldots, \frac{\Delta}{\Delta_N}\}$, and $x'(0) \in \mathbb{R}^N$. Our objective is to find K and α such that the synchronization error defined as:

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

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 $a \in \mathbb{R}^+$, $b \in \mathbb{R}$.

Therefore the problem we tackle in this paper can be formulated as follows. Determine K and α 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;
d) possibly some cost function (depending on K and α) is minimized.

For conditions a) and b) observe that, if we let $v := \begin{bmatrix} \mathbb{1}^\top & 0 \end{bmatrix}^\top$ and $w := \begin{bmatrix} 0 & \mathbb{1}^\top D^{-1} \end{bmatrix}^\top$ then we have that

$$(A_D - I)w = v \quad (A_D - I)v = 0$$

showing in this way that v and w are respectively an eigenvector and a generalized eigenvector of A_D in (II.5) associated with the eigenvalue 1. Notice moreover that $[\Omega \; 0]w = [\Omega \; 0]v = 0$, which shows that they are both unobservable.

Remark 2.1: The strategy proposed by Scardovi and Sepulchre in [9] for obtaining consensus for higher order systems is not applicable for time synchronization. Indeed for their method we have that in the non ideal case in which $D \neq I$, the eigenvalue 1 becomes observable and so we do not obtain consensus.

We need now to find methods able to give K and α satisfying conditions c) and d) of the previous list. Observe preliminarily that often the Δ_i's are slightly different and can be seen to be a small perturbation of the nominal value Δ. Therefore the dynamics of Eqn. (II.5) can be seen as the perturbation of the system where $D = I$. Concerning condition c), observe that, since the eigenvalues are continuous function of the matrix elements, if we find a solution satisfying condition c) letting $D = I$, this solution will continue to satisfy that condition even in case that D is a small enough perturbation of I. Similarly for condition d), if the cost function to be minimized is continuous in D, then the optimal solution found assuming $D = I$ will have approximately the same cost of the optimal solution obtained starting from the true D. For these reasons, in the stability analysis and in the optimization problems which will follow, we will assume that $D = I$. We will use the symbol A_I for the matrix A_D in (II.5) when $D = I$.

III. Stability analysis

The aim of this section is to study under which conditions on K and α the proposed synchronization algorithm yields $e(h)$ converging to zero; this is equivalent to require that $A_I := A_{D|D=I}$ in (II.5) has two eigenvalues in 1, as as seen before, while all the other eigenvalues belong to the open unit disc. For simplicity, in the sequel we will assume that K is symmetric. Let U be an orthogonal matrix formed with the eigenvectors of K, i.e., such that $U^*KU = \Lambda$ where $\Lambda = \operatorname{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ is the diagonal matrix containing the eigenvalues λ_i of K. Since $K\mathbb{1} = 0$, without loss of generality we will assume that $\lambda_1 = 0$.

Proposition 3.1: Consider the system (II.5) in the case where $D = I$. Then the remaining $2N-2$ eigenvalues of this system are inside the open unit disk if and only if

$$0 < \alpha < 1 \tag{III.1}$$

$$0 < \lambda_h < \frac{4}{2-\alpha}, \qquad 2 \le h \le N. \tag{III.2}$$

Proof: Using the change of variable $\bar{x}(h) := \operatorname{diag}\{U^*, U^*\}x(h)$ we see that the eigenvalues of A_I in (II.5) are the eigenvalues of the matrix

$$\begin{bmatrix} I-\Lambda & I \\ -\alpha\Lambda & I \end{bmatrix} \tag{III.3}$$

Its characteristic polynomial is $\prod_{i=1}^{N}(z-1)^2 + \lambda_i(z-1+\alpha)$. For $i = 1$ the previous polynomial gives the double eigenvalue 1. It remain to determine under which conditions the polynomials $(z-1)^2 + \lambda_i(z-1+\alpha)$, $i = 2, \dots, N$ are Shur-stable (i.e., have all roots inside the open unit disc). Using the bilinear transformation it can be seen that this happens if and only if conditions (III.1) and (III.2) hold true. ■

The previous proposition implies that K needs to be positive semidefinite. Assume now that we are given a graph $\mathcal{G} = (V, \mathcal{E})$ where $V = \{1, \dots, N\}$ and where $\mathcal{E} \subseteq V \times V$. Assume the $\mathcal{G}$ is undirected, namely that $(i,j) \in \mathcal{E}$ implies that $(i,j) \in \mathcal{E}$. This graph describes the communication topology, namely a node j can transmit information to the node i if and only if $(j,i) \in \mathcal{E}$. For this reason, we will say that a matrix K is compatible with $\mathcal{G}$ if and only if $K_{ij} \neq 0$ implies $(j,i) \in \mathcal{E}$. We will denote with $\mathcal{K}$ the set of rank $N-1$, symmetric, positive semidefinite matrices compatible with the given graph and such that $K\mathbb{1} = 0$. Without loss of generality we assume the eigenvalues $\lambda_1, \dots, \lambda_N$ are ordered such that

$$0 = \lambda_1 < \lambda_2 \le \dots \le \lambda_N.$$

One may wonder how difficult it is to construct a matrix K and to find α meeting the conditions of Proposition 3.1. The next proposition shows that, indeed, K and α guaranteeing stability can be constructed in a completely distributed way.

Proposition 3.2: Let $\mathcal{G}$ be a connected undirected graph and define P as the Metropolis stochastic matrix associated with the graph, i.e.,

$$P_{ij} = \begin{cases} \frac{1}{\max\{d_i, d_j\}+1} & \text{if } (i,j) \in \mathcal{E} \text{ and } i \neq j \\ 1 - \sum_{j \neq i} P_{ij} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{III.4}$$

where d_i denotes the number of neighbors of the node i, i.e., the cardinality of the set $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}, i \neq j\}$. Then $K := I - P$ and $\alpha = \frac{1}{2}$ meet the conditions of Proposition 3.1 thus guaranteeing stability.

Proof: Note that P is a stochastic symmetric matrix. By the Perron-Frobenius theorem, the conectivity of $\mathcal{G}$ implies that only one eigenvalue of P is 1 and the others are in the open interval $]-1, 1[$. Then the matrix $K := I - P$ and $\alpha = 1/2$ satisfy the second condition in Proposition 3.1. ■

Of course, other distributed strategies are possible for designing the matrix K, such as the Laplacian matrix [8].

IV. Convergence rate optimization

In the previous section we have seen that synchronization for a family of double integrators can be solved by properly choosing the matrix K and the parameter α, i.e., there exist K and α such that all systems, asymptotically, are synchronized. Of course one would like to go one step further asking whether it is possible to optimize K and α with respect to a specific performance index. In this section we consider the problem of obtaining K and α yielding the fastest convergence rate, which amounts to pushing the eigenvalues of the systems (II.5), i.e., the roots of $(z-1)^2 + \lambda_i(z-1+\alpha) = 0$, as close as possible to zero. Figure 1 displays a graphical representation of the root locus of $(z-1)^2 + \lambda(z-1+\alpha) = 0$ for $\alpha = 1/3$ and for λ that varies in the interval $[\lambda_2, \lambda_N] = [1,\ 1.9]$.

For small values of λ ($\lambda < 4\alpha$) the roots are complex conjugate, while for large λ ($\lambda > 4\alpha$) the roots are real. For $\lambda = 4\alpha$ there are 2 coincident roots in $z = 1 - \frac{\lambda}{2} = 1 - 2\alpha$. Optimizing for fastest convergence is equivalent to minimizing the absolute value of the largest eigenvalue in absolute value. We define

$$r(\lambda, \alpha) := \max\{|z| : (z-1)^2 + \lambda(z-1+\alpha) = 0\}, \tag{IV.1}$$

i.e., the maximum modulus of the two roots of the characteristic polynomial associated with the system (II.5). An explicit expression for $r(\lambda, \alpha)$ can be easily found. Indeed it is easy to see that

$$r(\lambda, \alpha) = \begin{cases} \sqrt{1 - \lambda + \alpha\lambda} & \text{if } \lambda < 4\alpha \\ \max\left\{ \left|1 - \lambda/2\left(1 \pm \sqrt{1 - 4\alpha/\lambda}\right)\right| \right\} & \text{if } \lambda > 4\alpha \end{cases}$$

As mentioned above, we have to minimize the largest of the $r(\lambda, \alpha)$'s as λ varies in $\sigma(K) \setminus 0$, where $\sigma(K)$ denotes the spectrum of K. Hence we define

$$R(K, \alpha) := \max_{\lambda \in \sigma(K) \setminus 0} r(\lambda, \alpha) \tag{IV.2}$$

The optimal values of α and K are the solution of the optimization problem

$$\{K_{opt}, \alpha_{opt}\} \in \operatorname*{arg\,min}_{K \in \mathcal{K},\ \alpha \in (0,1)} R(K, \alpha) \tag{IV.3}$$

where we recall that $\mathcal{K}$ is the set of rank $N-1$, symmetric, positive semidefinite matrices compatible with the given graph and such that $K\mathbb{1} = 0$. Being of rank $N-1$ is equivalent to the fact that $\lambda_2(K) > 0$. Notice that only matrices in $\mathcal{K}$ can yield stabilizing controllers and that, as observed in Proposition 3.2, such a set is nonempty if the supporting graph is connected. Define now $\mathcal{K}_N$ as the subset of $\mathcal{K}$ formed only by the matrices such that $\lambda_2(K) = 1$. Notice that $\mathcal{K} = \bigcup_{\beta > 0} \beta\mathcal{K}_N$. This implies that the previous optimization is equivalent to the following one

$$\{\bar{K}_{opt}, \alpha_{opt}, \beta_{opt}\} \in \operatorname*{arg\,min}_{\bar{K} \in \mathcal{K}_N,\ \alpha \in (0,1),\ \beta > 0} R(\beta\bar{K}, \alpha) \tag{IV.4}$$

since α_{opt} here is the same as in Eqn. (IV.3) and since $K_{opt} = \beta_{opt}\bar{K}_{opt}$. The optimization problem (IV.4) will now be considered.

Tedious but straightforward computations yield to the following results which are summarized as a lemma.

Lemma 4.1: The following facts hold true:

a) The function $r(\lambda, \alpha)$ is decreasing in λ for $\lambda < 4\alpha$ and it is increasing in λ for $\lambda > 4\alpha$.
b) The value of $R(K, \alpha)$ depends on K only through its smallest λ_2 and largest λ_N nonzero eigenvalues. More precisely we have

$$R(K, \alpha) \quad = \quad \max\{r(\lambda_2, \alpha), r(\lambda_N, \alpha)\} =: F(\alpha, \lambda_2, \lambda_N)$$

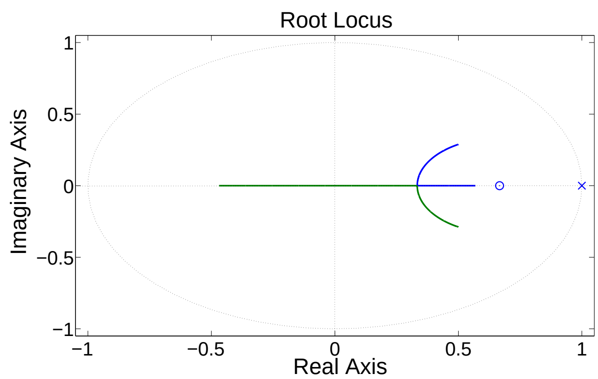

Fig. 1. Root locus of $(z-1)^2+\lambda(z-1+\alpha)=0$ for $\alpha=1/3$ as a function of $\lambda\in[\lambda_2,\lambda_N]$.

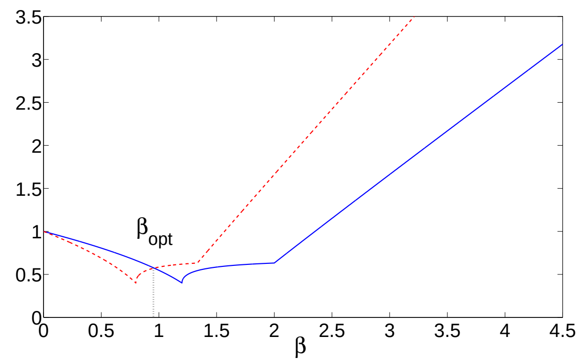

Fig. 2. The graphs of $r(\beta,\alpha)$ and of $r(\beta\lambda_N(\bar{K}_{opt}),\alpha)$ with $\alpha=0.3$ and $\lambda_N(\bar{K}_{opt})=1.5$

c) For fixed α and λ_2, the function $F(\alpha,\lambda_2,\lambda_N)$ is non-decreasing in λ_N.

Since, for any $\bar{K}\in\mathcal{K}_N$, $\alpha\in(0,1)$ and $\beta>0$ the condition $R(\beta\bar{K},\alpha)=F(\alpha,\beta,\beta\lambda_N(\bar{K}))$ holds, then a direct consequence of item (c) in the previous lemma is that $\bar{K}_{opt}$ in (IV.4) is given by

$$\bar{K}_{opt}\in\arg\min_{\bar{K}\in\mathcal{K}_N}\lambda_N(\bar{K}). \tag{IV.5}$$

Note that the set $\mathcal{K}_N$ is not convex but (IV.5) can be equivalently reformulated as the following convex optimization problem[1]

$$\bar{K}_{opt}\in\arg\min_{\bar{K}\in\mathcal{K},\ \lambda_2(\bar{K})\geq 1}\lambda_N(\bar{K}). \tag{IV.6}$$

Note that $\lambda_2(\bar{K}_{opt})=1$. If this was not true, i.e. $\lambda_2(\bar{K}_{opt})>1$, then the matrix $\bar{K}_{opt}^{'}:=\bar{K}_{opt}/\lambda_2(\bar{K}_{opt})$ would give a lower cost $\lambda_N(\bar{K}_{opt}^{'})=\lambda_N(\bar{K}_{opt})/\lambda_2(\bar{K}_{opt})<\lambda_N(\bar{K}_{opt})$ and $\lambda_2(\bar{K}_{opt}^{'})=1$, against the optimality assumption of $\bar{K}_{opt}$.

Thus the optimization problem (IV.4) is decoupled into the cascade of (IV.6) above followed by

$$\{\alpha_{opt},\beta_{opt}\}\in\arg\min_{\alpha\in(0,1),\beta>0}F(\alpha,\beta,\beta\lambda_N(\bar{K}_{opt})). \tag{IV.7}$$

Problem (IV.6) is a convex optimization problem since (i) the objective $\lambda_N(\bar{K})$ is a convex and symmetric spectral function (see e.g. [1]) and (ii) the set $\{\bar{K}\in\mathcal{K},\lambda_2(\bar{K})\geq 1\}$ is convex set. In particular (IV.6) can be formulated as a semi-definite program (SDP) for which standard and efficient software is available.

Once $\bar{K}_{opt}$ has been found, we need to find α_{opt} and β_{opt} solving (IV.7). To this aim, notice that

$$F(\alpha,\beta,\beta\lambda_N(\bar{K}_{opt}))=\max\{r(\beta,\alpha),r(\beta\lambda_N(\bar{K}_{opt}),\alpha)\}$$

and so, for fixed α and as a function of β, it is the maximum between $r(\beta,\alpha)$ and its stretched version $r(\beta\lambda_N(\bar{K}_{opt}),\alpha)$ (see Figure 2). For the properties of the function $r(\lambda,\alpha)$ mentioned in Lemma 4.1, this function is minimized for the unique β such that $r(\beta,\alpha)=r(\beta\lambda_N(\bar{K}_{opt}),\alpha)$ and so this equation gives $\beta_{opt}(\alpha)$ which can be found for instance using a bisection method. Finally, the optimal value of α can be found via a linear search for α ranging in the interval $(0,1)$.

V. Noisy model

In this section we will consider a noisy version of Eqn. (II.5). First of all, we allow for noisy measurements $y(h)$ of the form $x'(h)-y(h)=v(h)$ so that, inserting the control (II.4) in equations (II.2) and (II.3), a term of the form

$$\begin{bmatrix} K \\ \alpha K \end{bmatrix} v(h)$$

has to be added on the right hand side of (II.5). In addition we allow for a model noise to act on the second component of the state[2], thus obtaining

$$x(h+1)\ =\ A_I x(h)+\begin{bmatrix} K \\ \alpha K \end{bmatrix} v(h)+\begin{bmatrix} 0 \\ I \end{bmatrix} n(h), \tag{V.1}$$

where $v(h)$ and $n(h)$ are assumed to be uncorrelated white noises with zero mean and covariances $\mathbb{E}\left[n(t)n^\top(\tau)\right]=qI\delta(t-\tau)$ and $\mathbb{E}\left[v(t)v^*(\tau)\right]=rI\delta(t-\tau)$. We will also assume that, for all $h\geq 0$, both $n(h)$ and $v(h)$ are uncorrelated from the initial condition $x(0)$, which is assumed to be random. For clearness of exposition, a motivation of the model noise $n(h)$ in the clock synchronization problem will be given in Appendix A.

Clearly the presence of the noise prevents in general that $e(h)\to 0$. Therefore, in order to evaluate how much the performance of the algorithm degrades, we introduce the cost functional

$$J(K,\alpha):=\frac{1}{N}\limsup_{h\to\infty}\mathbb{E}\left[\|e(h)\|^2\right] \tag{V.2}$$

where the expectation is taken over the initial condition and the realizations of the noises. The cost J can be expressed as a function of α and of the eigenvalues λ_i as formalized in the following lemma.

Lemma 5.1: Let J be defined as in (V.2) and let $\lambda_1,\ldots,\lambda_N$ be the eigenvalues of K. Under the conditions (III.1) and (III.2) in Proposition 3.1 the cost $J(K,\alpha)$ satisfies:

$$J(K,\alpha)=\frac{1}{N}\sum_{h=2}^{N}\left(\frac{(\alpha^2-3\alpha+2)\lambda_h+2\alpha}{(1-\alpha)(4-(2-\alpha)\lambda_h)}r+ +\frac{(\alpha-1)\lambda_h+2}{\alpha(1-\alpha)(4-(2-\alpha)\lambda_h)\lambda_h^2}q\right) \tag{V.3}$$

Proof: As we have seen in Section II the system (II.5) has an unobservable component with two eigenvalues in 1. Under the assumptions of Proposition 3.1, all other eigenvalues are stable. Thus, restricting to the (stable) observable component of the state, the proof follows from a rather standard application of Ljapunov equations to compute the steady state covariance in linear state space models and is omitted in the interest of space. ■

[1] This optimization it can performed using the techniques proposed in [14].

[2] It should be observed that since we consider a double integrator, it would make little sense to add a model noise term also in the first component of the state.

The goal now is to design K and to choose α in order to minimize $J(K,\alpha)$, i.e. to solve

$$\underset{\alpha\in(0,1),\ K\in\mathcal{K}}{\arg\min}\ J(K,\alpha), \tag{V.4}$$

where $\mathcal{K}$ is the set of symmetric positive semidefinite matrices introduced in the previous section. Observe that, from the proof of Lemma 5.1 it follows that, if the conditions (III.1) and (III.2) are not satisfied, then $J(K,\alpha) := +\infty$. Now, given α, let

$$\mathcal{K}(\alpha) = \{K \in \mathcal{K} : \lambda_N < 4/(2-\alpha)\}. \tag{V.5}$$

In other words $\mathcal{K}(\alpha)$ is the set of positive semidefinite matrices compatible with the graph structure, satisfying $K\mathbb{1}=0$ and ensuring that, given α such that $0<\alpha<1$, the cost functional $J(K,\alpha)$ is finite. The minimization problem (V.4) can be treated in the following way. We start by observing that

$$\min_{\alpha\in(0,1),\ K\in\mathcal{K}} J(K,\alpha) = \min_{\alpha\in(0,1)} J\left(K^{opt}(\alpha),\alpha\right),$$

where $K^{opt}(\alpha) \in \arg\min_{K\in\mathcal{K}(\alpha)} J(K,\alpha)$. Assume now α fixed. We have the following proposition.

Proposition 5.2: Fix $\alpha \in (0,1)$ and let $\mathcal{K}(\alpha)$ be defined as in (V.5). Then the function $J(K,\alpha)$ defined on $\mathcal{K}(\alpha)$ is a convex function.

Proof: Consider the function $f:\mathcal{B}\to\mathbb{R}$ defined as

$$f(x_1,\dots,x_N) = \sum_{h=2}^{N}\left(\frac{(\alpha^2-3\alpha+2)x_h+2\alpha}{(1-\alpha)(4-(2-\alpha)x_h)}r + \frac{(\alpha-1)x_h+2}{\alpha(1-\alpha)(4-(2-\alpha)x_h)x_h^2}q\right) \tag{V.6}$$

where $\mathcal{B} = \left\{x\in\mathbb{R}^N : 0<x_i<4/(2-\alpha)\right\}$. We show that $f(x_1,\dots,x_N)$ is convex. To this aim consider the h-th term in the summation in the right-hand side of (V.6). Observe that $\frac{(\alpha^2-3\alpha+2)x_h+2\alpha}{(1-\alpha)(4-(2-\alpha)x_h)}$ is a convex hyperbola for $x_h \in\ ]0,4/(2-\alpha)[$. Moreover

$$\frac{(\alpha-1)x_h+2}{\alpha(1-\alpha)x_h^2(4-(2-\alpha)x_h)} = \frac{1}{\alpha(1-\alpha)}\left[\frac{\alpha}{8x_h}+\frac{1}{2x_h^2}+\frac{(2-\alpha)\alpha}{8(4-(2-\alpha)x_h)}\right]$$

where each term of the summation in the right-hand side of the above expression is convex for $x_h\in\ ]0,4/(2-\alpha)[$. Hence the function f is convex in $\mathcal{B}$. Finally observe that the function f is symmetric, i.e., it is invariant to any permutation of the vector entries x_h . Hence, it follows from the theory of convex spectral functions [1] that also J is a convex function. ■

From the above proposition, and from the fact that the set $\mathcal{K}(\alpha)$ is a convex set, it follows that the minimization problem

$$K^{opt}(\alpha) \in \underset{K\in\mathcal{K}(\alpha)}{\arg\min}\, J(K,\alpha) \tag{V.7}$$

is a convex optimization problem. In particular the solution of (V.7) can be obtained by suitable numerical algorithms. Once (V.7) has been solved, one is left with

$$\underset{\alpha\in(0,1)}{\arg\min}\, J\left(K^{opt}(\alpha),\alpha\right)$$

which reduces to a linear search over α and is easily solved.

VI. Numerical examples

In this section we provide two examples illustrating the approach proposed in this paper. Specifically, in Example 6.1 we simulate the algorithm described in Section II in a time-varying setup, while in Example 6.2 we deal with the minimization problems formulated in Section IV and in Section V.

Example 6.1: In Section II the convergence properties of the synchronization algorithm illustrated in Section II have been characterized, under the assumption that K is time-invariant. The goal of this example is to show the effectiveness of this synchronization algorithm also in a time-varying setup. To do so, we propose a comparison between the synchronous implementation given in (II.5) and a randomized asynchronous implementation, based on a broadcast communication protocol, that we describe next.

Let $\mathcal{G}$ be a generic undirected graph. To the graph $\mathcal{G}$ we can associate the doubly stochastic (Metropolis) matrix P_{Metr}, built as illustrated in Eqn. (III.4), and the corresponding matrix $K_{Metr} := I - P_{Metr}$. Assume now that at each time instant a node of $\mathcal{G}$ is randomly chosen with a probability $1/N$. Without loss of generality let i be the node chosen at time h and let us denote by $\mathcal{N}_i$ the set of its neighbors in $\mathcal{G}$. At time h, node i broadcasts the value $x_i'(h)$ to all the nodes in $\mathcal{N}_i$. The neighboring nodes $j\in\mathcal{N}_i$ update their state using the input

$$u_j(h) := [1\ \ \alpha]^T\, k_{ji}(x_j'(h)-x_i'(h)),$$

where k_{ji} denotes the element of the j-th row and i-th column of K_{Metr}. For all nodes ℓ which are not neighbors of node i, i.e. $\ell\notin\mathcal{N}_i$ we set $u_\ell(h)=0$. In more compact form the matrix $K(h)$ can be written as

$$K(h) = \sum_{j\in\mathcal{N}_i} k_{ji}\left(e_je_j^T - e_je_i^T\right). \tag{VI.1}$$

Observe that $\mathbb{E}[K(h)] = N^{-1}K_{Metr}$.

In Figure 3, we show the behavior of the synchronization error both for the synchronous implementation given in (II.5) and for the asynchronous implementation described above for a connected random geometric graph generated by choosing $N = 15$ points uniformly distributed in the unit square and by connecting with an edge each pair of points at distance less than 0.4. Notice that in the asynchronous implementation only one node transmits its information at each time instant. For this reason, in order to make a fair comparison, when analyzing the asynchronous implementation we sample the value of the synchronization error every N iterations of the algorithm. To be more precise, in Figure 3, we plot in dashed line the quantity $J_s(h) := \log\frac{1}{\sqrt{N}}\|e_s(h)\|$, where e_s denotes the synchronization error for the synchronous implementation, in solid line the quantity $J_a(h) := \log\frac{1}{\sqrt{N}}\|e_a(Nh)\|$, where e_a denotes the synchronization error for the asynchronous implementation. The synchronous algorithm in (II.5) has been implemented with $K = K_{Metr}$ and $\alpha=\alpha_{Metr}$, where $\alpha_{Metr} = \min_{\alpha\in(0,1)} R(K_{Metr},\alpha)$. In the asynchronous algorithm we built $K(h)$ as in (VI.1) and we fixed $\alpha = \alpha_{Metr}/N$. The factor $1/N$ in α used in the asynchronous implementation is related to the fact that the N asynchronous steps should be compared with one synchronous, as discussed above. As far as the initial condition is concerned, the speeds of the clocks (i.e., the elements of the matrix D) and the initial local time (i.e., the values $x_i'(0)$, $i\in\{1,\dots,N\}$) have been chosen randomly in the intervals $[0.9, 1.1]$ and $[0, 100]$, 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. The results obtained show the effectiveness of the randomized asynchronous implementation whose performance turns out to be comparable with the performance of the synchronous one.

Example 6.2: In this example we analyze numerically the optimization problems formulated in (IV.4) and in (V.4). To this aim, we introduce the following nomenclature. Given a connected undirected

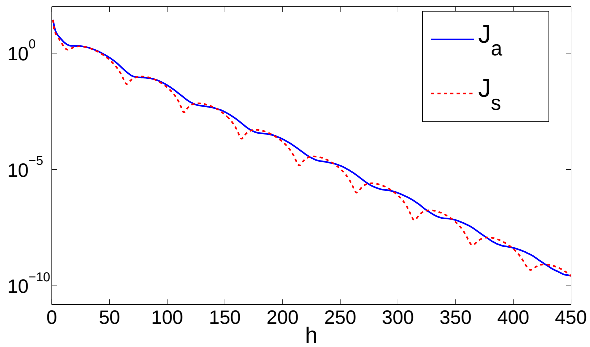

Fig. 3. Behavior of the synchronization error both for the synchronous implementation and asynchronous implementation.

graph $\mathcal{G}$, let

$$R_{Opt} = \min_{\substack{\alpha\in(0,1)\\ K\in\mathcal{K}}} R(K,\alpha), \quad J_{Opt} = \min_{\substack{\alpha\in(0,1)\\ K\in\mathcal{K}}} J(K,\alpha),$$

where R and J have been defined in (IV.2) and in (V.3), respectively. Now, we associate with the graph $\mathcal{G}$ the matrix P_{Metr}, built as illustrated in (III.4). Let $K_{Metr} := I - P_{Metr}$ and define

$$R_{Metr} := R(K_{Metr}, \alpha = 1/2)$$
$$J_{Metr} := J(K_{Metr}, \alpha = 1/2).$$

which correspond to the performance of the totally distributed design strategy proposed in Proposition 3.2. We run 30 experiments. At each experiment, we generated a connected random geometric graph, as in the previous example. For all the experiments we set the value of the parameters q and r, i.e., the variances of $n_i(t)$ and $v_i(t)$, equal to 1. We plotted the points (J_{Opt}, J_{Metr}) and the points $(\tau_{Opt}, \tau_{Metr})$ where $\tau_{Opt} = \frac{\log 0.05}{\log R_{Opt}}, \tau_{Metr} = \frac{\log 0.05}{\log R_{Metr}}$[3] in the upper figure and in the lower figure of Figure 4, respectively. In order to facilitate the comparison we also plot the bisector straight lines. As expected, in both cases, all the points lie above these lines. Moreover, observe that, in most cases, the improvement gained by choosing the optimal matrix K is considerable for both cost functionals.

VII. Conclusions

We have presented a second-order consensus algorithm for a family of non-identical double integrators which borrows tools from standard control theory and consensus algorithms. The main motivation comes from clock synchronization in a network of agents. The optimal controller for fastest rate of convergence in this class can be formulated as a convex optimization problem. Linearity also allows to perform a rather simple analysis of the effect of the noise on the asymptotic performance. While the analysis has been performed for a synchronous algorithm, numerical simulations show that an asynchronous implementation of the same algorithm has a comparable performance to the synchronous one. Indeed an important research direction is the theoretical analysis of asynchronous algorithms and their testing in real networks of clocks.

Appendix A

In the context of clock synchronization the process noise $n(h)$ in (V.1) can be justified as follows. Assuming that the clock's periods Δ_i are time varying functions, it is reasonable to write D in Eqn. (II.5) as

$$D(h) = I + S(h)$$

where $S(h) = \text{diag}\,\{S_i(h)\}_{i=1,..,N}$, $S_i(h) := \frac{\Delta - \Delta_i}{\Delta_i}$. It is reasonable to model the terms $S_i(h)$ as a random walk, i.e. $S_i(h) = S_i(h-1) + n_i(h)$. The noises $n_i(h)$ enters in the model in a multiplicative manner since $S_i(h)$ multiplies the second state component $x_i''(h)$. However x_i'' is always approximately equal to one[4] and hence it is reasonable to assume $S_i(h)x_i''(h) \simeq S_i(h)$. Under this approximation, and redefining the second component of the state as $x_i''(h) + S_i(h)$, we obtain the term

$$\begin{bmatrix} 0 \\ I \end{bmatrix} n(h)$$

on the right hand side of (V.1), where $n_i(h)$ is the i-th component of $n(h)$.

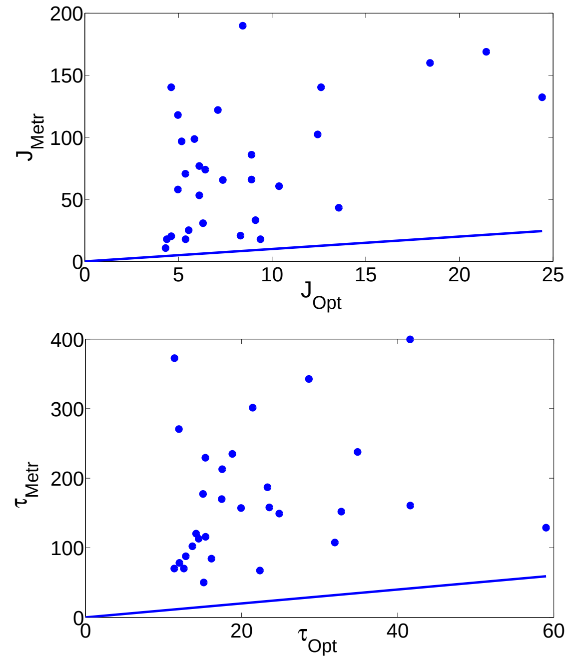

Fig. 4. Plot of the points (J_{Opt}, J_{Metr}) ***(upper)***. Plot of the points $(\tau_{Opt}, \tau_{Metr})$ ***(lower)***.

References

[1] J.-M. Borwein and A.-S. Lewis. *Convex Analysis and Nonlinear Optimazation*. CMS Books in Mathematics. Canadian Mathematical Society, 2000.

[2] 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)*, pages 147–163, 2002.

[3] R. Freeman, P. Yang, and K.M. Lynch. Stability and convergence properties of dynamic average consensus estimators. In *45th IEEE Conference on Decision and Control (CDC'06)*, pages 398–403, San Diego, December 2006.

[4] 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.

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

[3] Note that, given an positive real number ϵ, the quantity $\frac{\log \epsilon}{\log R_{Opt}}$ (respectively the quantity $\frac{\log \epsilon}{\log R_{Metr}}$) gives the asymptotic number of steps for the synchronization error decreasing of a factor ϵ when $K = K_{Opt}$ (respectively when $K = K_{Metr}$).

[4] In the practical implementation x_i'' is kept bounded in an interval of the form $[1-\epsilon, 1+\epsilon]$ to avoid dangerous drifts.

[6] Miklòs Maròti, Branislav Kusy, Gyula Simon, and Àkos Ldeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 39–49, 2004.

[7] W. Ren. On consensus algorithms for double-integrator dynamics. *IEEE Transactions on Automatic Control*, 53(6):1503–1509, 2008.

[8] R. Olfati Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in multi-agent networked systems. *Proceedings of IEEE*, 95(1):215–233, January 2007.

[9] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11):2557–2562, 2009.

[10] 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.

[11] 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)*, pages 2734–2739, San Diego, December 2006.

[12] D. Spanos, R. Olfati-Saber, and R.M. Murray. Dynamic consensus on mobile networks. In *Proceedings of the 16th IFAC World Congress*, July 2005.

[13] 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*, pages 1541–1546, 2008.

[14] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53:65–78, 2004.

[15] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 67(1):33–46, 2007.

[16] M. Zhu and S. Martinez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.