A PI Consensus Controller for Networked Clocks Synchronization

R. Carli [*] **A. Chiuso** [**] **L. Schenato** [*] **S. Zampieri** [*]

[*] *Department of Information Engineering, University of Padova,*
Padova, Italy, (emails: {carlirug,schenato,zampi}@dei.unipd.it).
[**] *Department of Management and Engineering, University of Padova,*
Vicenza, Italy, (email: chiuso@dei.unipd.it)

Abstract: In this paper we propose a novel distributed clock synchronization protocol for networks of clocks which have different initial offsets and internal clock speeds. The algorithm is based on an PI-like consensus protocol where the proportional (P) part compensates the different clock speeds while the integral part (I) eliminates the different clock offsets. This synchronization protocol is formally studied in its synchronous implementation and we provide both convergence guarantees as well optimal design using standard optimization tools when the underlaying communication graph is known. We also show how this protocol can be readily used to study the effect of noise and external disturbances on the steady-state performance. Finally, some simulations are presented.

1. INTRODUCTION

The extraordinary success of Internet and of wireless technologies has created the opportunity to interconnect hundreds to thousands of devices which can exchange information. This possibility to connect large numbers of devices which are physically distributed over large distances with a communication network offers the opportunity to accomplish new tasks and to control of the environment more effectively. However, large interconnected systems require coordination and cooperation to achieve these goals. One important problem to be solved in many applications involving a network of distributed devices is to maintain it temporally synchronized. These applications include, for example, mobile target tracking using a large number of motion detection devices Oh et al. (2005), power scheduling and TDMA communication schemes in wireless sensor networks Hohlt et al. (2004), and rapid synchronized coordination of powerlines nodes in electric power distribution networks for catastrophic power-outage prevention Amin and Schewe (2007).

Clock synchronization is an old problem and very sophisticated methodologies have been devised. The most naive strategy to synchronize two clock is simply elect one clock as a reference and then periodically update the offset of the other clock based on the clocks difference. However, if the two clocks have different speeds, the time difference between two clocks diverges in between synchronization updates, thus showing a saw-tooth evolution. A methodology to compensate for different time speeds is to estimate the relative speed and use it to correctly predict the reference node time. A very natural extension of the previous time synchronization strategy to a network of clocks is first to elect a root and to create a tree from the communication graph and then use the previous strategy where each son synchronizes itself with respect to its parent, see e.g. Ganeriwal et al. (2003), Maròti et al. (2004). Another very natural 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 synchronize itself with a another cluster-head, see Elson et al. (2002). Although these two strategies can be easily implemented and have shown remarkable performance Maròti et al. (2004) they suffer from two main problems. The first problem is robustness. In fact, if a node dies or a new node is added to the network 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. The second problem is that, depending on how the tree or clusters are built, it might happen that two clocks which are physically close and can communicate with each other belong to two different branches of the tree or two different clusters, thus possibly having large clock differences. This is particularly harmful in those applications such as TDMA communication which requires good synchronization of each node with its neighbors.

Another major problem in interconnected systems is the time delay between the clock readings of two distinct devices. In fact, the synchronization algorithms described above implicitly assume that the clock reading at the reference node and at synchronizing node is instantaneous. If this is not the case, due for example to software access time required to read the local clock, to propagation delay, and to the reading time at the receiver clock, then the synchronization performance can degrade and even generating instabilities in the algorithms. Although, there are algorithms that try to mitigate these effects, see e.g. Solis et al. (2006), the general trend is to solve this problem at hardware level by trying to read the clocks right before packet transmission and as soon the packet arrives, as

[⋆] This work has been supported in part by the national project *New techniques and applications of identification and adaptive control* funded by MIUR and by EU fund MIRC-6-CT-2005-014815 "SENSNET".

discussed in Maròti et al. (2004). Therefore, in this work we will neglect transmission delay and assume that clock reading are instantaneous.

In this paper, we propose a novel fully distributed time synchronization protocol based on a PI-like modification of standard consensus algorithms (see e.g. Olfati-Saber (2007)). The proportional part (P) of the consensus algorithm compensates for the different clock skews, while the integral part (I) compensate for the different clock offsets. The implementation of this protocol requires minimal computational and memory resources since each node needs only few operations. Recently, other fully distributed algorithms for clock synchronization have appeared. For example, the Reachback Firefly Algorithm (RFA) in Werner-Allen et al. (2005), a protocol inspired by the fireflies integrate-and-fire synchronization mechanism, is able to compensate for different clock offsets but not for different clock skews. On the opposite, the algorithm proposed in Simeone and Spagnolini (2007), adopting a control-based approach similar to this paper, is able to compensate for the clock skews but not for the offsets, i.e. at the end of the synchronization process, all clocks will measure a constant time difference with the other clocks. Distributed protocols that can compensate for both clock skews and offsets are the Tiny-Sync Protocol in Yoon et al. (2007), the Distributed Time-Sync Protocol in Solis et al. (2006) and the Average Time-Sync Protocols by Schenato and Gamba (2007). The first one is based on a type of robust linear regression, the second on distributed least-square estimator, and the last on a cascade of two consensus algorithms. They are all proved to synchronize a network of clocks in the absence of noise and delivery time-delay and they also show good performance in experimental testbeds. However it is difficult to mathematically predict the effect of noise on the steady-state performance, i.e. the distribution of synchronization errors. Differently, the synchronization protocol proposed in this paper can analyze not only the noiseless scenario but also the effect of noise in the steady-state performance. Moreover it allows for optimal design of protocol parameters if the graph of the network is known. It is important to remark that our protocol is proposed in a synchronous implementation, i.e. all nodes are supposed to communicate at the same time instant, while the other protocols mentioned above are all asynchronous, i.e. they do not need a specific coordination for the communication. We will come back to the consequence of this point in the conclusions.

The rest of the paper presents the mathematical formulation of the proposed protocol and also how the protocols parameters can be easily globally optimized through some standard numerical optimization tools. We also study the effect of noise in the overall system performance. We tested our protocol through some simulations, and finally we present some future research extensions.

2. PROBLEM FORMULATION AND THE PROPOSED SOLUTION

In this paper we will model a clock as a discrete time integrator $x(t+1) = x(t) + d$ where $x(t)$ denotes the local time at time t and $d > 0$ denotes the time drift (speed) of the clock. If we have a family of N clocks with local time $x_i(t)$, these will be in general characterized by different initial local time $x_i(0)$ and by different drifts d_i. We assume that it is possible to control each clock by a local input $u_i(t)$ as follows $x_i(t+1) = x_i(t) + d_i + u_i(t)$. By defining the N-dimensional vectors $x(t)$, $u(t)$ and d as vectors with components $x_i(t)$, $u_i(t)$ and d_i respectively, we can write the following vector model of the set of clocks

$$x(t+1) = x(t) + d + u(t).$$

Clocks synchronization, i.e. $x_i(t) = x_j(t)\ \forall i,j$, can be achieved by acting on the control input $u(t)$. In practice clock synchronization can only be achieved asymptotically, i.e. $\exists a, b \in \mathbb{R}$ such that

$$x(t) - (at+b)\mathbf{1} \overset{t\to\infty}{\longrightarrow} 0$$

where $\mathbf{1}$ denotes the column vector with all entries equal to 1. The control action $u_i(t)$ can only use *local* information. More precisely, the information exchange is described by a graph $\mathcal{G}$ with set of vertices $\{1, \dots, N\}$ in which we have an edge from j to i whenever the information coming from clock j can be used by $u_i(t)$ to control clock i. The problem we have to solve looks like a consensus problem and so the first attempt is to apply the standard linear consensus algorithm in which $u(t) = -Kx(t)$ where $K \in \mathbb{R}^{N\times N}$ is such that

(1) $I - K$ is an aperiodic and irreducible stochastic matrix.
(2) $K_{ij} \neq 0$ only if (j, i) is an edge of the graph $\mathcal{G}$.

It can be shown that this technique does not solve the problem since it is not able to get an agreement when both the drifts and the offsets (encoded by initial states $x_i(0)$) are different. This is intuitively motivated by the fact that, if one wants to control a system with a constant disturbance and obtain a zero asymptotic error, the internal model principle suggests that an integrator should be included in the controller. This intuition suggests the following structure

$$\begin{aligned} w(t+1) &= w(t) - Hx(t) \qquad w(0) = 0 \\ u(t) &= w(t) - Kx(t) \end{aligned}$$

where $w(t) \in \mathbb{R}^N$ is the controller state and $K, H \in \mathbb{R}^{N\times N}$. Observe that, in case the clocks are already synchronized, namely $d = a\mathbf{1}$ and $x(0) = b\mathbf{1}$, then we want that $u(t) = 0$ for all t. This equivalent to imposing that $K\mathbf{1} = H\mathbf{1} = 0$. In this paper we shall restrict to matrices H of the form $H = \alpha K$ and also assume that K is a symmetric matrix; this will allow to perform optimization of both α and K using standard tools in convex optimization. Symmetry can only be achieved only provided the associated graph is undirected. Note also that the controller for the i-th clock has the following form

$$\begin{aligned} w_i(t+1) &= w_i(t) - \alpha \sum_{j=1}^{N} K_{ij} x_j(t) \\ u_i(t) &= w_i(t) - \sum_{j=1}^{N} K_{ij} x_j(t). \end{aligned} \qquad (1)$$

Hence the global system will be the following

$$\begin{bmatrix} x(t+1) \\ w(t+1) \end{bmatrix} = \begin{bmatrix} I-K & I \\ -\alpha K & I \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}.$$

Remark 1. Note that, using the assumption $K\mathbf{1} = 0$ the following holds:

$$\sum_{j=1}^{N} K_{ij}x_j(t) = \sum_{j=1, j\neq i}^{N} K_{ij}(x_j(t) - x_i(t))$$

showing that this protocol, similarly to others in the literature, only needs the time differences between neighboring nodes. This is actually the only quantity which can reasonably be computed in a not-yet-synchronized network. For simplicity of exposition we shall however keep the notation in formula (1) in this paper.

3. STABILITY ANALYSIS

We first define

$$y(t) = (I - \frac{1}{N}\mathbf{1}\mathbf{1}^*)x(t)$$

It is clear that our objective is to drive $y(t)$ to zero. In order to show that the previous strategy solves the problem, observe that, since K is symmetric, then there exists a orthogonal matrix U such that $U^*KU = \Lambda$ where

$$\Lambda = \text{diag } \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$$

where $\lambda_i \in \mathbb{R}$. Observe that, from the assumption we introduced above that $K\mathbf{1} = 0$, we have that one of the λ's is zero and the associated column of U is $N^{-1/2}\mathbf{1}$. Moreover it follows that

$$U^T \left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^* \right) U = \text{diag } \{0, I_{N-1}\}. \qquad (2)$$

With no loss of generality we assume that $\lambda_1 = 0$ and that the first column od U is $N^{-1/2}\mathbf{1}$.

By defining the new vectors $\bar{x}(t) := U^T x(t)$, $\bar{w}(t) := U^T w(t)$, $\bar{d} := U^T d$, $\bar{y}(t) := U^T y(t)$. After this change of coordinates the model becomes

$$\begin{bmatrix} \bar{x}(t+1) \\ \bar{w}(t+1) \end{bmatrix} = \begin{bmatrix} I - \Lambda & I \\ -\alpha\Lambda & I \end{bmatrix} \begin{bmatrix} \bar{x}(t) \\ \bar{w}(t) \end{bmatrix} + \begin{bmatrix} \bar{d} \\ 0 \end{bmatrix}$$

The overall system consists in N decoupled 2-dimensional systems

$$\begin{bmatrix} \bar{x}_h(t+1) \\ \bar{w}_h(t+1) \end{bmatrix} = \begin{bmatrix} 1 - \lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_h(t) \\ \bar{w}_h(t) \end{bmatrix} + \begin{bmatrix} \bar{d}_h \\ 0 \end{bmatrix} \qquad (3)$$

¿From (2) it follows that $\bar{y}_1(t) = 0$ and $\bar{y}_h(t) = \bar{x}_h(t)$ for $h \neq 1$. If all the systems with $h \neq 1$ are asymptotically stable, then the steady state solution must satisfy the equation

$$\begin{bmatrix} \bar{x}_h(\infty) \\ \bar{w}_h(\infty) \end{bmatrix} = \begin{bmatrix} 1 - \lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_h(\infty) \\ \bar{w}_h(\infty) \end{bmatrix} + \begin{bmatrix} \bar{d}_h \\ 0 \end{bmatrix}$$

which has unique solution $\bar{x}_h(\infty) = 0$ and $\bar{w}_h(\infty) = -\bar{d}_h$. This shows that $\bar{y}(t)$ tends to zero and so $y(t)$ tends to zero as well. More precisely, since

$$\bar{x}_1(t) = \bar{d}_1 t + \bar{x}_1(0)$$

and since $\bar{x}_h(t)$ converges to zero if $h \neq 1$, then

$$x(t) = U\bar{x}(t) \longrightarrow U \begin{bmatrix} \bar{d}_1 t + \bar{x}_1(0) \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \frac{1}{N}\mathbf{1}\mathbf{1}^*(dt + x(0))$$

which is exactly what we require.

We now want to find conditions on K and α ensuring the stability of the systems (3) for all $h \neq 1$. The characteristic polynomial of these systems is

$$(z-1)^2 + \lambda_h(z - 1 + \alpha)$$

Using standard techniques (such as the root locus method) it is easy to verify that the previous polynomial has both roots inside the unit circle if and only if $0 < \alpha < 1$ and $0 < \lambda_h < 4/(2-\alpha)$. Therefore there is no loss of generality if we further restrict K to be also positive semidefinite. In particular we shall denote with $\mathcal{K}$ the set of symmetric, positive semidefinite matrices compatible with the graph structure and such that $K\mathbf{1} = 0$. Under these conditions the non zero eigenvalues of this matrix are all positive. Without loss of generality we assume the eigenvalues $\lambda_1, .., \lambda_N$ are ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N.$$

Remark 2. A solution can be found for instance by taking the Laplacian matrix L of the undirected graph. This is a symmetric matrix with eigenvalues belonging to the interval $[0, 2]$ and such that $L\mathbf{1} = 0$. If the graph is strongly connected, the L has only one eigenvalue equal to zero. Then by taking $K = \beta L$ with any β such that $0 < \beta < 2/(2-\alpha)$ we have a solution of the clock synchronization problem.

4. CONTROLLER OPTIMIZATION

In the previous Section we have seen that the clock synchronization problem can be solved by properly choosing the matrix K and the parameter α, i.e. there exist K and α such that all clocks, asymptotically, are synchronized. Of course one would like to go one step further asking whether it is possible to optimize K and α for fastest convergence, which amounts to pushing the eigenvalues of the systems (3), i.e. the roots of $(z-1)^2 + \lambda(z-1+\alpha) = 0$, as close as possible to zero. The root locus of $(z-1)^2+\lambda(z-1+\alpha) = 0$ is shown in figure 1 for $\alpha = 1/3$ as λ 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 $z = 1 - \frac{\lambda}{2} = 1 - 2\alpha$. Since the initial conditions of the clocks may be arbitrary, optimizing for fastest convergence is equivalent to minimizing the absolute value of the largest (in modulus) eigenvalue. Therefore, for future use, we define

$$r(\lambda, \alpha) := \max\{|z| : (z-1)^2 + \lambda(z-1+\alpha) = 0\}, \qquad (4)$$

i.e. the maximum modulus of the roots of the characteristic polynomial associated to the systems (3). An explicit expression for $r(\lambda, \alpha)$ can be easily found. When the roots are complex conjugate, i.e. for $\lambda < 4\alpha$, $r(\lambda, \alpha) = \sqrt{1 - \lambda + \alpha\lambda}$. Instead, when the roots are real, i.e. for $\lambda > 4\alpha$, $r(\lambda, \alpha) = \max\{|1 - \lambda/2 \left(1 \pm \sqrt{1 - 4\alpha/\lambda}\right)|\}$. These simple expressions are useful to derive analytically several statements which, for reasons of space, we only discuss graphically with the help of the root locus in fig. 1. As mentioned above we have to minimize the largest (as $\lambda \in \sigma(K)$) of the $r(\lambda, \alpha)$'s, which we define as

$$R(K, \alpha) := \max_{\lambda \in \sigma(K)} r(\lambda, \alpha) \qquad (5)$$

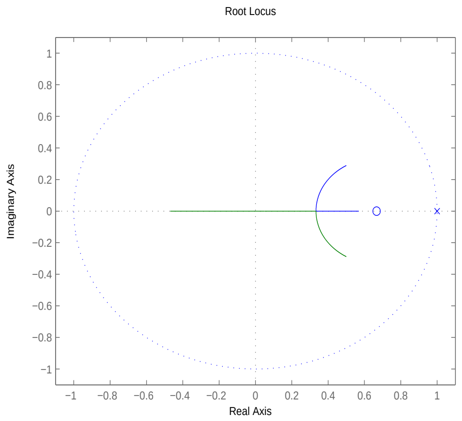

Fig. 1. Root locus of $(z-1)^2+\lambda(z-1+\alpha)=0$ for $\alpha=1/3$ as a function of $\lambda\in[\lambda_2,\lambda_N]$.

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

$$\{K_{opt},\alpha_{opt}\}\in \underset{K\in\mathcal{K},\ \alpha\in[0,1]}{\arg\min}\ R(K,\alpha). \tag{6}$$

Note that (see fig. 1) $r(\lambda,\alpha)\le\max\{r(\lambda_2,\alpha),r(\lambda_N,\alpha)\}$; therefore $R(K,\alpha)$ depends on K only through its maximum and minimum eigenvalues $\lambda_2:=\lambda_2(K)$ and $\lambda_N:=\lambda_N(K)$, i.e.

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

It is also clear that, for fixed values of λ_2 and λ_N, the optimal choice of α is such that $\frac{\lambda_2}{4}\le\alpha_{opt}\le\frac{\lambda_N}{4}$, i.e. the roots corresponding to λ_2 are complex conjugate while those corresponding to λ_N are real. In fact, assume $\alpha_{opt}>\frac{\lambda_N}{4}$; all roots would be complex conjugate; in such case the absolute value of the roots associated to λ is $\sqrt{1-\lambda+\alpha\lambda}$, which is monotonically increasing in α. Therefore, by decreasing α one would reduce the absolute value of all roots (and hence in particular of the maximum one), against the optimality assumption. A similar argument holds if one would assume that $\alpha_{opt}<\frac{\lambda_2}{4}$. This observation will be useful later on. For reasons which will become clear later on it is convenient to introduce a new parametrization of the matrix K by letting $K:=\beta K'$, $\beta>0$ where $\lambda_2(K')=1$. Note that, if the matrix K is symmetric, positive semidefinite and compatible with the graph structure, also K' is so. Note also that, if $\lambda'\in\sigma(K')$, then $\lambda:=\beta\lambda'\in\sigma(K)$, so that in particular $\lambda_2(K)=\beta$ and $\lambda_N(K)=\beta\lambda_N(K')$. Consider now the new costs

$$\begin{aligned} r'(\lambda',\alpha,\beta) &:= r(\beta\lambda',\alpha)\\ R'(K',\alpha,\beta) &:= R(\beta K',\alpha)\end{aligned} \tag{7}$$

Define also $\lambda'_N:=\lambda_N(K')$ and

$$\hat{R}'(\lambda'_N,\alpha,\beta):=\hat{R}(\beta,\beta\lambda'_N,\alpha) \tag{8}$$

Therefore, by letting $K_{opt}=\beta_{opt}K'_{opt}$, the solution of the optimization problem (6) can be obtained from the solution $\{K'_{opt},\alpha_{opt},\beta_{opt}\}$ of

$$\underset{\substack{K'\in\mathcal{K}\\ s.t.\lambda_2(K')=1}\ \ \alpha\in[0,1],\ \beta>0}{\arg\min}\ \hat{R}'(\lambda'_N,\alpha,\beta). \tag{9}$$

Note that the optimal value of the cost depends on K' only through its maximum eigenvalue λ'_N.

The dependence of the cost $\hat{R}'(\lambda'_N,\alpha_{opt},\beta_{opt})$ on λ'_N is particularly simple. In fact, having found the values of α_{opt} and β_{opt} corresponding to a given λ'_N, as shown above $\frac{\lambda_2}{4}=\frac{\beta_{opt}}{4}\le\alpha_{opt}\le\frac{\beta_{opt}\lambda'_N}{4}=\frac{\lambda_N}{4}$ hold true, i.e. $(z-1)^2+\lambda_2(z-1+\alpha_{opt})=0$ has complex roots while $(z-1)^2+\lambda_N(z-1+\alpha_{opt})=0$ has real roots. It thus follows that $r'(\lambda'_N,\alpha_{opt},\beta_{opt})$ is locally increasing in λ'_N. Therefore also $\hat{R}'(\lambda'_N,\alpha_{opt},\beta_{opt})$ is locally increasing in λ'_N. It follows that one needs first to optimize K' so that λ'_N is minimized, i.e.

$$K'_{opt}\in\underset{\substack{K'\in\mathcal{K}\\ s.t.\lambda_2(K')=1}}{\arg\min}\ \lambda'_N; \tag{10}$$

Note that optimization problem above can be equivalently be formulated as

$$K'_{opt}\in\underset{\substack{K'\in\mathcal{K}\\ s.t.\lambda_2(K')\ge1}}{\arg\min}\ \lambda'_N, \tag{11}$$

in which both the cost function and the optimization set are convex. Thus the optimization problem (9) is decoupled into the cascade of (11) above followed by

$$\{\alpha_{opt},\beta_{opt}\}\in\underset{\alpha\in[0,1],\beta>0}{\arg\min}\ \hat{R}'(\lambda_N(K'_{opt}),\alpha,\beta). \tag{12}$$

Problem (11) is a convex optimization problem since (i) the objective λ'_N is a convex and symmetric spectral function (see e.g. Borwein and Lewis (2000)) and (ii) the set of positive semidefinite matrices K' compatible with the graph structure, such that $K'\mathbf{1}=0$ and $\lambda_2(K)\ge1$ is a convex set. In particular (11) can be formulated as a SDP program for which standard and efficient software is available.

We shall hence regard (11) as solved and consider only problem (12). The solution turns out to be particularly simple and reduces to a standard linear search over the two parameters α and β thanks to the following observation: for each value of $\alpha\in[0,1]$, there is a set of values of β which are candidates for making α the optimal value for the given matrix K'_{opt}. As discussed above[1] $\frac{\beta}{4}\le\alpha_{opt}\le\frac{\beta\lambda'_N}{4}$ must hold, so that $\frac{4\alpha}{\lambda'_N}\le\beta\le4\alpha$ is the range of values we seek for. For the given value of α, $r'(1,\alpha,\beta)$ is monotonically decreasing in $\beta\in[\frac{4\alpha}{\lambda'_N},4\alpha]$ while $r'(\lambda'_N,\alpha,\beta)$ is monotonically increasing in $\beta\in[\frac{4\alpha}{\lambda'_N},4\alpha]$. Therefore the optimal value $\beta_{opt}(\alpha)$ for given α is the unique value $\beta_{opt}(\alpha)$ such that $r'(1,\alpha,\beta_{opt}(\alpha))=r'(\lambda'_N,\alpha,\beta_{opt}(\alpha))$ and can be found for instance using a bisection method.

Last but not least, the optimal value of α can be found via a linear search for α ranging in the interval $[0,1]$. It turns out (see figure 2, but we haven't been able to prove it) that $\hat{R}'(\lambda'_N,\alpha,\beta_{opt}(\alpha))$ is convex in α, making the search even simpler, e.g. using again a bisection method.

[1] Recall that $\lambda'_2=1$.

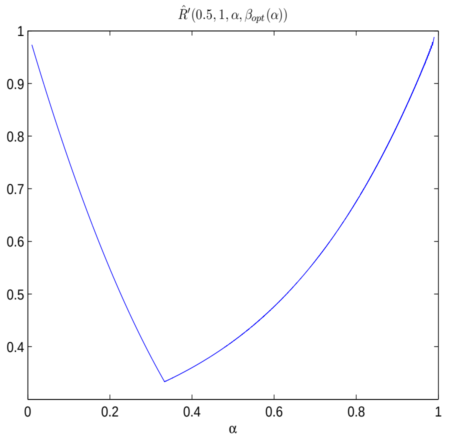

Fig. 2. Cost function $\hat{R}'(\lambda'_N, \alpha, \beta_{opt}(\alpha))$ for $\lambda'_N = 2$.

5. NOISY MODEL

The model and the controller can be generalized in order to consider the fact that the drifts d_i can be time varying and the fact that the $x_j(t)$ received by the clock i from the clock j will be corrupted by noise (for instance quantization noise). More precisely we model the drift of the i-th clock and the information that the i-th clock sends to its neighbors, at the time instant t, respectively as $d_i + n_i(t)$ and $x_i(t) + v_i(t)$, where $n_i(t)$ and $v_i(t)$ are zero mean noises such that $\mathbb{E}\left[n_i^2(t)\right] = q$, $\forall\, i$, $\mathbb{E}\left[v_i^2(t)\right] = r$, $\forall\, i$, and $\mathbb{E}\left[n_i(t)n_j(\tau)\right] = 0$ and $\mathbb{E}\left[v_i(t)v_j(\tau)\right] = 0$ if $i \neq j$ or $t \neq \tau$. Furthermore we assume that $\mathbb{E}\left[n_i(t)v_j^T(\tau)\right] = 0$ for each pair of clocks i, j and time instants t, τ. We can define the N-dimensional vectors $v(t)$ and $n(t)$ as vectors with components $v_i(t)$ and $n_i(t)$, i.e. $v(t) = [v_1(t), \ldots, v_N(t)]^T$ and $n(t) = [n_1(t), \ldots, n_N(t)]^T$. Clearly $\mathbb{E}\left[n(t)n^T(\tau)\right] = q\delta(t-\tau)I$ and $\mathbb{E}\left[v(t)v^T(\tau)\right] = r\delta(t-\tau)I$. More compactly we have that the model describing these undesired phenomena can be written as

$$\begin{bmatrix} x(t+1) \\ w(t+1) \end{bmatrix} = \begin{bmatrix} I-K & I \\ -\alpha K & I \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} -K \\ -\alpha K \end{bmatrix} v(t) + \\ + \begin{bmatrix} I \\ 0 \end{bmatrix} d + \begin{bmatrix} I \\ 0 \end{bmatrix} n(t)$$

We consider again the variables $y, \bar{y}$. Moreover we define the new variables $z := w + d$, $\bar{z} := U^T z$, $\bar{v} := U^T v$ and $\bar{n} := U^T n$. After this change of coordinates and after some manipulations the model becomes

$$\begin{bmatrix} \bar{y}(t+1) \\ \bar{z}(t+1) \end{bmatrix} = \begin{bmatrix} I-\Lambda & U^T\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^*\right)U \\ -\alpha\Lambda & I \end{bmatrix} \begin{bmatrix} \bar{y}(t) \\ \bar{z}(t) \end{bmatrix} + \\ + \begin{bmatrix} -\Lambda \\ -\alpha\Lambda \end{bmatrix} \bar{v}(t) + \begin{bmatrix} U^T\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^*\right)U \\ 0 \end{bmatrix} \bar{n}(t)$$

¿From (2) it follows that, as for the noiseless model, the overall system continues to consist of N decoupled systems which are the following

$$\begin{bmatrix} \bar{y}_1(t+1) \\ \bar{z}_1(t+1) \end{bmatrix} = I \begin{bmatrix} \bar{y}_1(t) \\ \bar{z}_1(t) \end{bmatrix}$$

and

$$\begin{bmatrix} \bar{y}_h(t+1) \\ \bar{z}_h(t+1) \end{bmatrix} = \begin{bmatrix} 1-\lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix} \begin{bmatrix} \bar{y}_h(t) \\ \bar{z}_h(t) \end{bmatrix} + \begin{bmatrix} -\lambda_h \\ -\alpha\lambda_h \end{bmatrix} \bar{v}_h + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \bar{n}_h$$

for $2 \leq h \leq N$. We recall that $\bar{y}_1(t) = 0$, $\forall\, t$. In order to analyze the asymptotic property of the noisy model it is convenient to introduce the N matrices

$$P_h(t) := \mathbb{E}\left\{ \begin{bmatrix} \bar{y}_h(t) \\ \bar{z}_h(t) \end{bmatrix} \left[\bar{y}_h(t) \;\; \bar{z}_h(t)\right] \right\}$$

After some calculations we obtain the following N recursive equations

$$P_1(t+1) = P_1(t)$$

and

$$P_h(t+1) = \begin{bmatrix} 1-\lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix} P_h(t) \begin{bmatrix} 1-\lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix}^T + \\ + \lambda_h^2 \begin{bmatrix} 1 & \alpha \\ \alpha & \alpha^2 \end{bmatrix} r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} q$$

for $2 \leq h \leq N$. Clearly the presence of the noises prevents in general that $y(t) \to 0$. In order to evaluate how much the performance of our algorithm degrades we introduce the following functional cost $J(\alpha, K) := \frac{1}{N}\mathbb{E}\left[\|y(\infty)\|^2\right]$. After some calculations one can show that

$$J(\alpha, K) = \frac{1}{N} \sum_{h=1}^{N} P_h^{(1,1)}(\infty)$$

where $P_h^{(1,1)}(\infty)$ denotes the element in the first row and first column of the matrix $P_h(\infty)$. Since $\bar{y}_1(t) = 0$, $\forall\, t$, we have that $P_1^{(1,1)}(t) = 0$, $\forall\, t$. Instead for $2 \leq h \leq N$, reasoning as in Section 3, if all the systems are asymptotically stable, we have that the steady state solution $P_h(\infty)$ must satisfy the following equation

$$P_h(\infty) = \begin{bmatrix} 1-\lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix} P_h(\infty) \begin{bmatrix} 1-\lambda_h & 1 \\ -\alpha\lambda_h & 1 \end{bmatrix}^T + \\ + \lambda_h^2 \begin{bmatrix} 1 & \alpha \\ \alpha & \alpha^2 \end{bmatrix} r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} q$$

from which it follows after some calculations that

$$P_h^{(1,1)}(\infty) = -\frac{2r\lambda_h^2 + 2q - 3r\alpha\lambda_h^2 + 2\lambda_h r\alpha + r\alpha^2\lambda_h^2}{\lambda_h\left(-4 + 2\lambda_h + 4\alpha - 3\alpha\lambda_h + \alpha^2\lambda_h\right)}$$

Hence it results that

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

Now we would like to design K and to choose α in order to minimize the above quantity. We restrict to the following set of matrices

$$\mathcal{K}' = \{K \in \mathcal{K} : \lambda_N < 2/(4-\alpha)\}$$

where $\mathcal{K}$ is the set of matrices introduced in the previous section. In other words $\mathcal{K}'$ is the set of positive semidefinite matrices compatible with the graph structure, satisfying $K\mathbf{1} = 0$ and ensuring that the all the terms $P_h^{(1,1)}(\infty)$, $2 \leq h \leq N$, are finite. Hence our goal is to solve

$$\arg\min_{K \in \mathcal{K}',\ \alpha \in (0,1)} J(\alpha, K). \qquad (13)$$

This above minimization problem can be treated in the following way. We start by observing that

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

where $K^{opt}(\alpha) \in \arg\min_{K \in \mathcal{K}'} J(\alpha, K)$. Assume now α fixed. One can show that the h-th term of the summation in the right-hand side of (13) is a convex function in $\lambda_h \in \left(0, \frac{2}{4-\alpha}\right)$. Hence, similarly to (11), it follows that

$$K^{opt}(\alpha) \in \underset{K \in \mathcal{K}'}{\arg\min}\, J(\alpha, K) \tag{14}$$

is a convex optimization problem since (i) the objective $J(\alpha, K)$ is a convex and symmetric function spectral function and (ii) the set $\mathcal{K}'$ is a convex set. Therefore the solution of (14) can be performed by suitable numerical algorithms. Hence we can consider (14) as solved and analyze the following problem

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

As for (12) the solution of the above problem turns out to be particularly simple and reduces to a linear search over the parameter α.

6. NUMERICAL EXAMPLES

In this section we provide an example illustrating the algorithm proposed in this paper. In this simulation we consider a connected random geometric graph generated by choosing $N = 15$ points uniformly distributed in the unit square, and then placing an edge between each pair of points at distance less than 0.3. The speeds of the clocks and the initial local time has been chosen randomly respectively inside the interval $[0.5, 1.5]$ and the interval $[0, 200]$. In Figure 3 is depicted the behavior of $x(t)$.

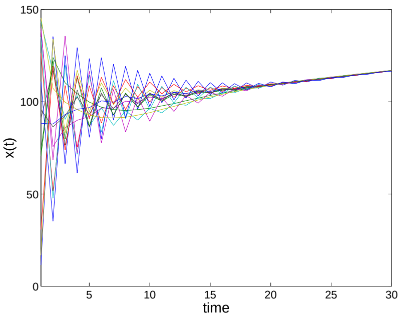

Fig. 3. Behavior of x(t)

7. CONCLUSIONS

We have presented a linear synchronization protocol which borrows tools from standard control theory and consensus algorithms. The optimal (in terms of speed of convergence) controller in the class can be formulated as a convex optimization problem. Linearity also allows to perform a rather simple analysis of the effect of noise on the asymptotic performance. This is to be regarded as one of the major contributions w.r.t. previous works in which nonlinear schemes have been proposed; in fact the nonlinearity makes it very hard to study the effect of noise. Last but not least, we should mention that the algorithm discussed here is, in contrast with previous works, synchronous. As mentioned in the Section 2, the algorithm only requires time differences between adjacent nodes; however, when the network is not synchronized, these time differences cannot be computed synchronously. The effect of this jitter can indeed be lumped into an additive error term, which, thanks to the linearity of the scheme, can be analyzed using standard tools. A more thorough analysis of this and related aspects, together with an in-depth comparison with existing approaches, is of course part of our plans for the future.

REFERENCES

M Amin and P.F. Schewe. Preventing blackouts. *Scientific American*, pages 60–67, May 2007.

J.M. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization, Theory and Examples.* Canadian Mathematical Society Books in Mathematics. Spring-Verlag, New York, 2000.

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.

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.

B. Hohlt, L. Doherty, and E. Brewer. Flexible power scheduling for sensor networks. In *IEEE and ACM International Symposium on Information Processing in Sensor Networks (IPSN'04)*, April 2004.

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

S. Oh, L. Schenato, and S.S. Sastry. A hierarchical multiple-target tracking algorithm for sensor networks. In *IEEE International Conference on Robotics and Automation*, pages 2197 – 2202, Barcelona, Spain, April 2005.

R. Olfati-Saber. Algebraic connectivity ratio of ramanujan graphs. In *Proceedings of the 2007 American Control Conference*, July 2007.

L. Schenato and G. Gamba. A distributed consensus protocol for clock synchronization in wireless sensor network. In *to appear in Proceedings of IEEE Conference on Decision and Control*, 2007.

O. Simeone and U. Spagnolini. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP Journal on Wireless Communications and Networking*, 2007:Article ID 57054, 13 pages, 2007. doi:10.1155/2007/57054.

R. Solis, V. Borkar, and P. R. Kumar. A new distributed time synchronization protocol for multihop wireless networks. In *To appear in 45th IEEE Conference on Decision and Control (CDC'06)*, San Diego, December 2006.

Geoff Werner-Allen, Geetika Tewari, Ankit Patel, Matt Welsh, and Radhika Nagpal. Firefly-inspired sensor network synchronicity with realistic radio effects. In *ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, November 2005.

Suyoung Yoon, Chanchai Veerarittiphan, and Mihail L. Sichitiu. Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Journal of Sensor Networks*, 3(2), 2007.