How to synchronize biological clocks

G. Russo* and M. di Bernardo*†

Abstract

This paper is concerned with a novel algorithm to study networks of biological clocks. A new set of conditions is established that can be used to verify whether an existing network synchronizes or to give guidelines to construct a new synthetic network of biological oscillators that synchronize. The methodology uses the so-called contraction theory from dynamical system theory and Gershgorin disk theorem. The strategy is validated on two examples: a model of glycolisis in yeast cells and a synthetic network of repressilators that synchronizes.

Keywords: Synthetic Biology, Networks, Synchronization

1 Introduction

In biology, much research effort has been focused on the analysis and investigation of synchronization of biological rhythms. The most famous example is that of circadian rhythms in mammals, regulated by an endogenous biological clock entrained by external signals from the environment (Gonze et al. 2005). The environmental light-dark cycle for example acts as one of the most important pacemakers. Moreover, this particular clock is a robust system in the sense that it can be entrained to follow daily variations such as temperature and light (Bernard et al. 2003), (Kunz & Achermann 2003). Another important example of synchronization and coordination of biological clocks is the cell cycle through which cells periodically duplicate their genome and divide (Tyson et al. 2002). Understanding the emergence and coordination of rhythmic phenomena regulating the activities of living organisms requires the investigation of the cooperative behavior leading to synchronization.

Over the past few years, several attempts have been made not only at analysing but also at devising strategies to synchronize biological networks. Seminal work in this area can be found in (Winfree 2001, 2nd Edition), where the problem was simplified by considering only phase variations between the cells of a generic biological network. Many researchers have studied synchronization in genetic networks both experimentally (Yamaguchi et al. 2003), (Pye 1969), and theoretically, (McMillen et al. 2002), (You et al. 2004), (Kiss et al. 2007), (Mirollo & Strogatz 1990).

*The authors are with the Department of Systems and Computer Engineering, University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy. E-mail:{giovanni.russo2,mario.dibernardo}@unina.it.
†Also with the Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, U.K.
As novel robust synthetic oscillators become readily available in the literature (see for example (Chilov & Fussenegger 2004)), a pressing open problem is to identify effective methods to construct networks of oscillators that synchronize. Most of the proposed methods in the literature rely on the construction of Lyapunov functions and numerical resolution of linear matrix inequalities (LMIs) (Li et al. 2006), (Li et al. 2007) or on the use of perturbation methods (Bier et al. 2000). While theoretically sound, these approaches can be difficult to implement in practice. Also, only few papers take explicitly into account the presence of noise and/or parameter mismatches.

The aim of this paper is to address this urgent open problem and propose a novel, general, algorithm to analyse synchronization in existing biological networks or to construct synthetic networks which synchronize. The algorithm is based on a rigorous mathematical analysis and can be used to investigate well-known synchronization mechanisms such as quorum sensing. The key step is the use of the recent theory of contracting dynamical systems ((Lohmiller & Slotine 1998), (Jouffroy & Slotine 2004), (Slotine et al. 2004), (Pham et al. n.d.)) in the context of networks of biological oscillators and the application of Gershgorin disk theorem from matrix theory (Horn & Johnson 2001). By using these analytical tools, we establish a novel set of sufficient conditions for synchronization that are then validated on a set of representative examples illustrating how the algorithm can be successfully applied in order to engineer ad hoc biological clocks that synchronize or to check if an already existing network can synchronize. For the sake of brevity, we expound here the main results and their application, leaving to the Supplementary Information the proofs and the mathematical details.

2 Results

2.1 Studying synchronization in biological systems

Models of biological systems are often described by expressing the rate of change of their current state as a function of the actual state. This is formalized using ordinary differential equations of the form:

\[ \dot{x} = f(x), \]

where \( x \in \mathbb{R}^m \) is the vector of the state variables of the system (gene product or protein levels), and \( f : \mathbb{R}^m \to \mathbb{R}^m \) is the law that associates to the state \( x(t) \), representing the level of gene products and proteins of a biological system, their rate of production, \( \dot{x}(t) \).

The flow, \( \phi \), is defined as the function of time \( t \) and initial value \( x_0 \) which represents the set of solutions of (1) (Ott 1993). Thus, \( \phi(x_0,t) \) is the evolution of the concentrations of gene products and proteins of the system starting from their initial level, \( x_0 \).

A region, \( C \), in state space, i.e. the space defined by the state variables, is said to be a contraction region for (1), if any infinitesimal length converges exponentially to zero under the action of the system flow. By path integration, this immediately implies that the length of any finite path in \( C \) converges exponentially to zero. In (Lohmiller & Slotine 1998), it is shown that the contracting property of a region...
Definition 1. A region in state space, say \( C \), is called a contraction region for (1), if its Jacobian, \( J = \frac{\partial f}{\partial x} \), is uniformly negative definite in \( C \). Namely, we define \( C \) as:

\[
C := \left\{ x \in \mathbb{R}^m : \frac{1}{2} \left( \frac{\partial f}{\partial x} + \frac{\partial f^T}{\partial x} \right) \leq -\beta I, \beta > 0, \forall t \in \mathbb{R}^+ \right\}.
\] (2)

If \( C \) agrees with the entire set \( \mathbb{R}^m \), then (1) is called contracting.

A notable result also obtained in (Lohmiller & Slotine 1998) and reported in Section 1 of the Supplementary Information is the so called contraction principle, which states that trajectories generated in a contraction region do not escape from it and converge to each other. The rate at which the two trajectories converge is called contraction rate.

The key stage in the application of contraction theory to synchronization of networks of oscillators is the construction of the so called virtual system. The virtual system (Jouffroy & Slotine 2004) is defined as the system which has the trajectories of the nodes (and thus the evolutions of gene products and proteins of each biological oscillator) as particular solutions. Formally, the virtual system depends on the state variables of the oscillators and on some virtual state variables. The substitution of the \( i \)-th node state variables into the virtual state variables returns the dynamics of the \( i \)-th node of the network. The proof of the contracting property with respect to these virtual state variables immediately implies synchronization, as shown in the next example.

Example 1. Consider two nonlinear systems coupled in the following manner:

\[
\dot{z} = f(z) + h(w) - h(z),
\] (3)

\[
\dot{w} = f(w) + h(z) - h(w),
\] (4)

where \( h \) is some output function of the system states through which they are coupled. A suitable virtual system can be chosen as

\[
\dot{x} = f(x) - 2h(x) + h(z) + h(w) := \varphi(x, z, w).
\] (5)

Note, as described above, that such a system depends on the state variables of the nodes of the network, i.e. \( z \) and \( w \), and on the virtual state variables, i.e. \( x \). Furthermore, substituting \( z \) and \( w \) for \( x \) in (5), gives (3) and (4) respectively. Thus, the trajectories of the nodes are particular solutions (say \( x \)-solutions) of the virtual system, indeed

\[
\varphi(z, z, w) = f(z) + h(w) - h(z),
\]

\[
\varphi(w, z, w) = f(w) + h(z) - h(w).
\] (6)

Recall that the contracting property of the virtual system with respect to the virtual state variables implies that all its \( x \)-solutions converge to each other. Thus, if the virtual system is contracting the two particular \( x \)-solutions \( z \) and \( w \) will converge to each other. Synchronization is then attained.
This, in turn, implies that to prove synchronization of (3) and (4), it will suffice to show that (5) is contracting with respect to the $x$ state variable. In so doing, differentiation of (5) with respect to $x$ has to be performed, yielding

$$\delta \dot{x} = \left( \frac{\partial f(x)}{\partial x} - 2 \frac{\partial h(x)}{\partial x} \right) \delta x.$$  \hspace{1cm} (7)

Contractivity of the virtual system can then be ensured by making the symmetric part of the Jacobian matrix $\frac{\partial f(x)}{\partial x} - 2 \frac{\partial h(x)}{\partial x}$ uniformly negative definite, e.g. by imposing the negativity of all the eigenvalues of the system Jacobian.

Also, contraction theory can be extended to a stochastic framework, (Pham et al. n.d.). Consider a noisy system described by

$$\dot{x} = f(x) + \sigma(x,t) \xi_t,$$  \hspace{1cm} (8)

where $f: \mathbb{R}^m \to \mathbb{R}^m$, $\sigma(x,t): \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^{md}$ being a smooth matrix valued function, $\xi_t$ representing $d$ channels containing white noise (Oksendal & Bernt 2003). In what follows we assume that solution to (8) exists and is unique.

In (Pham et al. n.d.), it is proved that if (1) is contracting, then the mean distance between any two trajectories of (8) is upper bounded by a quantity that is directly proportional to the contraction rate of the noise-free system and inversely proportional to the maximum of the noise variance. Basically, this result, also reported in Section 1.1 of the Supplementary Information, states that any contracting system automatically rejects white noise. Thus, if the noise-free virtual system is contracting, then when noise is present, so-called stochastic synchronization will be achieved, where all trajectories converge towards some bounded set in phase space (whose dimension depends on the noise level and contraction rate).

Remarks

- As shown in Section 2.2, the construction of a virtual system similar to that of Example 1 is necessary for the application of the proposed algorithmic procedure. It is then important to find conditions ensuring that a virtual system exists for a given network. In (Russo & di Bernardo n.d.) it is shown that for networks with general topology it is always possible to associate an appropriate virtual system. Note that the Jacobian can also be used to investigate the structural stability of a biological system as shown in (Prill et al. 2005);

- Even if a virtual system exists for generic networks there is not a general procedure for its construction. In the examples of Section 2.2 the virtual systems are constructed by direct inspection. However, examples for the construction of the virtual system associated to different networks, recurrent in engineering and biological applications, can be found in (Slotine et al. 2004), (Pham et al. n.d.), (Wang & Slotine 2005);

- The structure of the connections between nodes strongly influences the algebraic form of the virtual system: discussions regarding this point can be found e.g. in (Gerard & Slotine 2006), (Botero &
Slotine 2006), (Chung & Slotine 2007), (Russo & di Bernardo n.d.). In the examples of the present paper, the networks considered have an all to all topology. It is interesting to note that the topology of the network is reflected into the structure of the virtual system. In Example 1, the topology is indeed responsible of the term $2h(x)$ appearing in (5):

- One of the main features of contraction theory is that it does not require the knowledge of a specific attractor to perform stability analysis: this implies that oscillations are not really important for the analysis of synchronization. Hence, nodes with non stationary aperiodic behavior could be synchronized using the proposed methodology. Analysis of synchronization of non periodic orbits can be found, for example, in (Botero & Slotine 2006).

2.2 An algorithm to synchronize biological clocks

We present an algorithm for the construction of a synthetic biological circuit that spontaneously synchronizes. Obviously, the same algorithm can be used to check if an already existing biological network can self-synchronize. The algorithm gives constraints on the topology of the network and the system parameters yielding a set of conditions for synchronization (CFS) that need to be fulfilled for the spontaneous emergence of synchronous behavior in the network of interest.

If some noise and parameter mismatches are present in the network, the application of the algorithm on the nominal system provides constraints ensuring the achievement of the stochastic synchronization regime. By nominal system, we refer to either the noise free version of the system, or the system having parameter values averaged over all nodes of the network. The fact that the algorithm takes explicitly into account noise and parameter mismatches is particularly useful in biology, where those sources of uncertainty are relevant. Moreover, from a mathematical point of view, the conditions provided by the algorithm are sufficient. In particular, if they are satisfied the network will certainly synchronize.

2.2.1 Initialization

Given a network of interest, the algorithm starts with the derivation of the network equations and the construction of an appropriate virtual system, as described in Example 1. The next step is the computation of the Jacobian matrix, say $J$, of the virtual system. The elements $J(i,j)$ of $J$ are the partial derivatives of the vector field $\varphi(x)$, i.e. $J(i,j) = \frac{\partial \varphi_i}{\partial x_j}, \forall i, j : 1, ..., m$. Namely, we have

$$J := \begin{bmatrix}
J(1,1) & J(1,2) & \ldots & J(1,m) \\
J(2,1) & J(2,2) & \ldots & J(2,m) \\
\vdots & \vdots & \ddots & \vdots \\
J(m,1) & \ldots & J(m, m-1) & J(m,m)
\end{bmatrix}. \tag{9}$$

By construction, the $j$-th element on the $i$-th row of $J$, $J(i,j)$, being non-zero indicates that species $j$ regulates species $i$. Thus, each row of $J$ represents the set of regulations on the species associated to that row. Moreover, the elements $J(i,j)$ is associated to the rate of production of the species (gene product or
protein) \( i \) with respect to the level of species (gene product or protein) \( j \). The diagonal elements of \( J \), i.e. \( J(i,i) \), represent the rate of production of the species \( i \) with respect to itself; in the rest of the paper they will be termed as self degradation rates. Note that the elements \( J(i,j) \) are in general functions of all the other state variables, i.e. the rate of production of species \( i \) with respect to species \( j \) could be influenced by the concentrations of other species in the system. In (9) the presence of null elements on the \( i \)-th row of \( J \) is not excluded, indicating that some genes or proteins of the system do not regulate the gene or protein \( i \); the number of zero elements on the \( i \)-th row of \( J \) is denoted by \( n_{0i} \).

### 2.2.2 Next steps

Once the algorithm has been initialized, the next stage generates a set of constraints on the system parameters, topology and coupling strength that, if satisfied, guarantee synchronization. In particular, for each row of \( J \), the next step consists in checking the truth or falsity of the following statements:

**S1:** \( J(i,i) < 0 \),

**S2:** \( J(i,i) = -\beta_i, \quad 0 < \beta_i < +\infty \),

**S3:** if \( J(i,j) \neq 0 \), then \( J(j,i) = 0 \).

From a biological viewpoint, S1 implies that the self-degradation rate of the \( i \)-th species in the synthetic circuit is uniformly negative, S2 states that the self degradation rate of the \( i \)-th species in the synthetic circuit is equal to some negative constant, \(-\beta_i\), and S3 states that if species \( i \) regulates species \( j \), then the reverse regulation cannot occur.

The algorithm consists of two nested loops with the aim of exploring all rows (outer loop) and columns (inner loop) of the virtual Jacobian \( J \) (a schematic flow diagram of the algorithm can be found in Section 3 of the Supplementary Information). The outcome is a set of inequalities [or conditions for synchronization (CFS)] that need to be satisfied (or verified) in order to guarantee that the network of interest synchronizes.

Starting with the first row, the first step is checking whether the diagonal element \( J(i,i) \) fulfills condition S1. If this is not the case, then the inequality corresponding to the fulfillment of S1 is added to the list of CFS generated by the algorithm. Then, S2 is checked on the same element and all other elements of that row are scanned to establish whether condition S3 is verified. For each element, \( J(i,j), i \neq j \), there are four possibilities:

- \( J(i,i) \) fulfills S2 and \( J(i,j) \) fulfills S3 (indicated as \( S1 - S2 - S3 \) in what follows and in Table 1);
- \( J(i,i) \) fulfills S2 and \( J(i,j) \) does not fulfill S3 (\( S1 - S2 - S\overline{3} \));
- \( J(i,i) \) does not fulfill S2 and \( J(i,j) \) fulfills S3 (\( S1 - S\overline{2} - S3 \));
- \( J(i,i) \) does not fulfill S2 and \( J(i,j) \) does not fulfill S3 (\( S1 - S\overline{2} - S\overline{3} \)).
To each of the cases listed above, the algorithm associates a corresponding inequality (see Table 1), which is added to the CFS generated by the algorithm. Note that in the special cases where the same property holds for all Jacobian elements, as explained in the Remarks of Section 3 of the Supplementary Information, the inequalities in Table 1 can be simplified yielding those given in Table 2.

Once all rows and columns have been scanned, the algorithm stops and a list of CFS is obtained. Note that, if the aim is to verify whether an existing network of interest synchronizes, it suffices that just one condition is not satisfied to conclude that the network is not guaranteed to synchronize. Alternatively, if the aim is to construct a biological network that synchronizes then the CFS generated by the algorithm can be used as design guidelines. Indeed, if all CFS are verified then synchronization will certainly be achieved.

All the inequalities involved with the application of the algorithm presented above have a clear biological interpretation. In particular, their fulfillment points towards a balance between the production and degradation rates of each species in the system as explained in Section 3.3 of the Supplementary Information. To gain a better understanding of the biological implications of each condition, we apply the algorithm to the following examples.

The first example is the analysis of an already existing biological clock, the Glycolysis process in yeast cells, showing how the algorithm can be used to predict the onset of synchronization.

Then, we move to the problem of making a synthetic network of Repressilators coupled using the quorum-sensing mechanism synchronize. By means of the algorithm outlined so far, we determine the biochemical parameters that must be chosen in order to achieve the desired synchronous regime.

2.2.3 Example 1: glycolysis in yeast cells

Glycolysis is the step-by-step breakdown of glucose and the storing of the released Gibbs energy in the form of ATP (Voet & Voet 2004). It has been observed that individual yeast cells in a suspension can synchronize their concentrations to get in phase with each other (Chen et al. 2005). In (Bier et al. 2000), a model for glycolysis of Saccharomyces cerevisiae is given and synchronization among different cells is proved analytically using a perturbation method. We now apply our algorithm to this model in order to verify that synchronization can indeed be attained. As we will see, our method is constructive. Specifically, it explicitly provides an estimate of the network parameters (e.g. the coupling strength between the nodes) and the biochemical parameters of each cell, required to achieve synchronization.

The mathematical model used in (Bier et al. 2000) (and schematically represented in Figure 5 of the Supplementary Information) is

\[
\begin{align*}
\dot{G}_1 &= V_{in} - k_1 G_1 T_1, \\
\dot{T}_1 &= 2k_1 G_1 T_1 - k_p \frac{T_1}{T_{km} + T_1} + \varepsilon (T_2 - T_1), \\
\dot{G}_2 &= V_{in} - k_1 G_2 T_2, \\
\dot{T}_2 &= 2k_1 G_2 T_2 - k_p \frac{T_2}{T_{km} + T_2} + \varepsilon (T_1 - T_2).
\end{align*}
\]
Here \( G \) stands for glucose and \( T \) for ATP. \( V_{in} \) represents the constant inflow of glucose, \( k_1 \) represents the phospho-fructokinase activity. Moreover, ATP is also broken down: this process is modeled by a Michaelis-Menten kinetics term which has \( k_m \) and \( k_p \) as its activation threshold and maximum inhibition rate respectively. When \( \varepsilon = 0 \) the two systems are uncoupled and each follows its own limit cycle. The coupling term, \( \varepsilon \), among yeast cells is greater than zero. Physically, this means that when \( T_2 > T_1 \), the production of \( T_1 \) is speeded up while the production of \( T_2 \) is slowed down. The coupling terms reflect the fact that there is a diffusive mass transfer between the two systems.

The virtual system corresponding to the two-cell network in (10) is chosen, by direct inspection, to be

\[
\dot{G} = V_{in} - k_1 G T,
\]

\[
\dot{T} = 2k_1 G T - k_p \frac{T}{k_m + T} - 2\varepsilon T + \varepsilon T_1 + \varepsilon T_2.
\]

Indeed, if \( G_1, T_1 \) and \( G_2, T_2 \) are substituted for \( G, T \) in (11) the dynamics of the first and the second cell are returned respectively.

Differentiation of (11) with respect to \( x := [G, T] \) yields the virtual Jacobian matrix

\[
J = \begin{bmatrix}
-k_1 T & -k_1 G \\
2k_1 T & 2k_1 G - 2\varepsilon - k_p k_m \frac{1}{(k_m + T)^2}
\end{bmatrix}.
\]

The algorithm is then started by checking if condition S1 is satisfied for the diagonal elements. Clearly, this condition is satisfied for \( J (1,1) \) thanks to the positiveness of \( k_1 \) and \( T \). However, \( J (2,2) \) does not satisfy S1. Thus, an explicit algebraic condition has to be given on this element, resulting in the first CFS generated by the algorithm. Namely, we must have \( J(2,2) < 0 \), i.e.

\[
2k_1 G - 2\varepsilon - k_p k_m \frac{1}{(k_m + T)^2} < 0.
\]

The algorithm then proceeds by checking if S2 is satisfied, i.e. \( J (i,i) = -\beta_i \). For the system of interest this condition is violated for the two diagonal elements of the virtual Jacobian. Moreover, the off-diagonal elements of \( J \) are all different from zero, resulting in a violation of S3. Thus, the diagonal and off-diagonal elements do not fulfill S2 and S3 respectively and the corresponding CFS are given in the last row of Table 2. In this case, the virtual system is contracting (and thus synchronization is achieved) if positive real parameters, \( p_1, p_2 \), can be found such that the following set of inequalities is consistent:

\[
k_1 T > \frac{p_2}{p_1} k_1 G,
\]

\[
-2k_1 G + 2\varepsilon + k_p k_m \frac{1}{(k_m + T)^2} > \frac{p_1}{p_2} 2k_1 T.
\]

As all elements of the virtual Jacobian have been scanned the algorithm then stops with the CFS given by (13),(14) and (15) having been generated. The CFS can now be used to find constraints on the system parameters as follows. The first necessary step is to find a region in the parameters space satisfying (13). Note that the term \( 2k_1 G \) is positive definite, while the term \( -2\varepsilon - k_p k_m \frac{1}{(k_m + T)^2} \) is negative definite. Thus, to satisfy (13) we need to make the negative definite term dominant with respect to \( 2k_1 G \). This can be
done for example by choosing $k_1$ much smaller than $k_p k_m$, in accordance with the parameters used in (Bier et al. 2000) where $k_1 = 0.02$, $k_m = 13$ and $k_p = 6$. The advantage for this choice is that (13) can be satisfied without using $\varepsilon$. This parameter can then be tuned to satisfy (14) and (15), which can be written as

$$\frac{p_2}{p_1} G < T < \frac{p_2}{p_1} \left( -G + \frac{\varepsilon}{k_1} + \frac{k_p k_m}{2k_1 (k_m + T)^2} \right),$$

(16)

and thus

$$G < \frac{p_1}{p_2} T < \psi(G),$$

(17)

with $\psi(G) := \left( -G + \frac{\varepsilon}{k_1} + \frac{k_p}{2k_1 k_m} \right)$. Since $p_1$ and $p_2$ are arbitrary positive real numbers, we can use them to rescale $T$ so that it will be enclosed in the region in phase space bounded by $G$ and $\psi(G)$. Clearly, this can be done only if $\psi(G) > G$.

Now, let’s term $G_{MAX}$ the maximum possible value of $G$ (this can be established numerically or from experiments). Then, if $k_p k_m / (2k_1) >> G_{MAX}$ and $\varepsilon >> k_1$, we obtain $\psi(G) > G$. For example, choosing $k_1 = 0.01$, $\varepsilon = 0.2$, $k_p = 5$, $k_m = 13$, all CFS given by the algorithm remain satisfied and, as shown in Figure 1, a stable synchronous state is indeed attained.

Note that the CFS provided by the algorithm require that the maximum break down rate of ATP, $k_p$, and its activation coefficient, $k_m$, are larger than the strength of the phospho-fructokinase activity. At the same time, this strength must be lower than the coupling strength between the cells given by $\varepsilon$.

### 2.2.4 Example 2: network of Repressilators

The Repressilator is a synthetic biological circuit of three genes inhibiting each other in a cyclic way (Elowitz & Leibler 2000). As shown in Figure 2, gene $lacI$ (associated to the state-variable $c_i$ in our model) expresses protein $LacI$ ($C_i$), which inhibits transcription of gene $tetR$ ($a_i$). This translates into protein $TetR$ ($A_i$), which inhibits transcription of gene $cI$ ($b_i$). Finally, the protein $CI$ ($B_i$) translated from $cI$ inhibits expression of $lacI$, completing the cycle.

In (Garcia-Ojalvo et al. 2004), (Wang et al. 2006), (Zhou et al. 2007) a modular addition to the classical Repressilator circuit is proposed with the aim of coupling different oscillators using the quorum sensing mechanism.

Quorum sensing is the process by which many bacteria coordinate gene expression according to the local density of signaling molecules produced by other bacteria. It provides a broadcast strategy for the exchange of information between bacteria. Namely, we could think of bacteria as nodes in a network that becomes fully connected via an all-to-all topology when quorum sensing is present.

In (Garcia-Ojalvo et al. 2004), it is shown that in the case of infinite cell dilution, the system consists of a population of uncoupled non-identical oscillators, thus synchronization cannot be achieved. As the cell density increases, quorum sensing provides a mechanism of intercell coupling, which leads to partial frequency locking of the cells. In (Zhou et al. 2007) and (Wang et al. 2006) it is shown that an impulsive
control law with period close to an integer multiplier of the averaged network period can induce/enhance the emergence of collective behavior.

The coupling circuit, incorporated in each cell is shown in Figure 2. It is obtained by placing the gene that encodes LuxI under the control of gene LacI belonging to the Repressilator. LuxI synthesizes a small molecule, AI, known as autoinducer (labeled as $S_i$ in our model), which can diffuse freely through the cell membrane. When AI binds to a second protein, LuxR, the resulting complex activates transcription of an additional copy of LacI (in the coupling circuit) which inhibits tetR in the Repressilator.

We will use the example described above as a viable representative case to test our algorithm to ensure synchronization. Our results are in accordance with those reported in (Li et al. 2006), (Li et al. 2007) where numerical methods based on linear matrix inequalities are used to prove the emergence of synchronous behavior. To model the dynamics of gene expression in the cell, one must keep track of the temporal evolution of all mRNA and protein concentrations. Note that, for the sake of simplicity, variations in the cell density are neglected here.

The resulting mathematical model of the $i$-th oscillator in the network is

\[
\begin{align*}
\dot{a}_i &= -a_i + \frac{\alpha}{1+c_i^\alpha} \\
\dot{b}_i &= -b_i + \frac{\alpha}{1+b_i^\alpha} \\
\dot{c}_i &= -c_i + \frac{\alpha}{1+c_i^\alpha} + \frac{ks_i}{1+S_i} \\
\dot{A}_i &= (\beta + \Delta \beta_i) (a_i - A_i) \\
\dot{B}_i &= (\beta + \Delta \beta_i) (b_i - B_i) \\
\dot{C}_i &= (\beta + \Delta \beta_i) (c_i - C_i) \\
\dot{S}_i &= -k_{s0}S_i + k_{s1}A_i - \eta (S_i - S_e) \\
\dot{S}_e &= -k_{se}S_e + \eta_{ext} \sum_{j=1}^{N} (S_j - S_e) = -k_{se}S_e + k_{diff} (\bar{S} - S_e)
\end{align*}
\]

having chosen the Hill coefficient equal to 2 as in (Garcia-Ojalvo et al. 2004). Note here the presence of the term $\Delta \beta_i$ that models, as a representative example, the possible mismatches among the Repressilator circuits. In particular, referring to (Garcia-Ojalvo et al. 2004) we set the parameter $\beta$ equal to 2 while $\Delta \beta_i$ is chosen from a Gaussian distribution with zero mean and standard deviation equal to 5.

The first step for the application of the algorithm, as explained in Section 2.2.1, is to derive an appropriate virtual system describing the noise free network. Note that, in this case, the noise free version of (18) is immediately obtained by imposing $\Delta \beta_i = 0$. A good choice for the virtual system was then found to be
\[ \begin{align*}
\dot{a} &= -a + \frac{a}{1 + C^2} \\
\dot{b} &= -b + \frac{a}{1 + A^2} \\
\dot{c} &= -c + \frac{a}{1 + B^2} + (KS_i)/(1 + S_i) \\
\dot{A} &= \beta(a - A) \\
\dot{B} &= \beta(b - B) \\
\dot{C} &= \beta(c - C) \\
\dot{S} &= -K_{s0}S + K_{s1}A - \eta(S - S_e) \\
\dot{S}_e &= -K_{se}S_e + \eta_{ext}(S_1 + \ldots + S_N) - \eta_{ext}N S_e
\end{align*} \]  

Indeed, by direct inspection it is easy to check that substituting the state variables of the nodes dynamics to the virtual variables, i.e. \([a_i, b_i, c_i, A_i, B_i, C_i, S_i, S_e]\) for \([a, b, c, A, B, C, S, S_e]\), the noise free dynamics of the \(i\)-th Repressilator circuit is obtained. In this way the virtual system encompasses the solutions of each noise free oscillator in the network as one of its particular solutions as required by the algorithm.

Differentiation of (19) yields the Jacobian matrix

\[
J = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & f_1(C) & 0 & 0 \\
0 & -1 & 0 & f_1(A) & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & f_1(B) & f_2(C) & 0 & 0 \\
\beta & 0 & 0 & -\beta & 0 & 0 & 0 & 0 \\
0 & \beta & 0 & 0 & -\beta & 0 & 0 & 0 \\
0 & 0 & \beta & 0 & 0 & -\beta & 0 & 0 \\
0 & 0 & 0 & k_{s1} & 0 & 0 & -k_{s0} - \eta & \eta \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -k_{se} - k_{diff}
\end{bmatrix}
\]  

(20)

where \(f_1\) and \(f_2\) denote the partial derivatives of decreasing and increasing Hill functions with respect to the state variable of interest and \(k_{diff} = \eta_{ext}N\). Note that matrix \(J\) has the form of a hierarchy (see Section 1 of Supplementary Informations for more details). Thus, since \(J(8,8)\) (associated to the all-to-all coupling between oscillators due to \(S_e\)) is negative definite, the algorithm can be applied to the submatrix \(\tilde{J}\) obtained by neglecting the last row and column of \(J\) and corresponding to the Repressilator circuit, i.e.

\[
\tilde{J} = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & f_1(C) & 0 & 0 \\
0 & -1 & 0 & f_1(A) & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & f_1(B) & f_2(C) & 0 & 0 \\
\beta & 0 & 0 & -\beta & 0 & 0 & 0 & 0 \\
0 & \beta & 0 & 0 & -\beta & 0 & 0 & 0 \\
0 & 0 & \beta & 0 & 0 & -\beta & 0 & 0 \\
0 & 0 & 0 & k_{s1} & 0 & 0 & -k_{s0} - \eta
\end{bmatrix}
\]  

(21)

Once the algorithm has been initialized, it is necessary to check if S1 is satisfied. In this case, (21) shows that \(\tilde{J}\) has negative constant diagonal elements. Hence, both S1 and S2 are satisfied for all the diagonal
elements. Moreover, it is easy to verify that all off-diagonal elements of $\tilde{J}$ fulfill S3. In this case, as shown in Table 2, one CFS is generated by the algorithm. This condition implies that the self degradation rate of at least one species of the system must be stronger than the production rate due to its interaction with the other species. Unfortunately, this condition is not satisfied by the network as it is and, accordingly, while synchronization might be possible, there is no guarantee that a synchronous evolution will be emerge.

Therefore, application of the algorithm presented in this paper suggests that to guarantee synchronization the network equations must be somehow modified. The easiest way to satisfy the CFS generated by the algorithm is to make the self degradation rate of one of the proteins stronger than its production rate, since those equations are linear and depend only on two state variables, i.e. the protein and the corresponding mRNA from which it is translated. Thus, it suffices to modify just one differential equation of (18) to ensure synchronization. Without loss of generality, we focus, for instance, on the equation modelling the protein TetR concentration, changing it into

$$\dot{A} = \gamma a - \beta A,$$

where $\gamma$ represents the new value of the protein translation rate chosen so that $\gamma < \beta$ in order for the CFS to remain satisfied.

Remark

It is important to note that since parameter mismatches are present among the Repressilators, under these CFS stochastic synchronization is guaranteed. In particular, note that a virtual system for the noisy network can be written as (8) with

$$f(x) = \begin{bmatrix} -a + \alpha / (1 + C^2) \\ -b + \alpha / (1 + A^2) \\ -c + \alpha / (1 + B^2) + (Ks_i) / (1 + S_i) \\ \beta (a - A) \\ \beta (b - B) \\ \beta (c - C) \\ -Ks0S + Ks1A - \eta (S - S_e) \\ -KseS_e + \eta_{ext} (S_1 + ... + S_N) - \eta_{ext} NS_e \end{bmatrix},$$

(23)
\[
\sigma(x,t) = \begin{bmatrix}
0 \\
0 \\
0 \\
(a - A) \\
(b - B) \\
(c - C) \\
0 \\
0
\end{bmatrix},
\]
and \(\xi_t = \Delta \beta\), where \(\Delta \beta\) is a white noise term encompassing the various fluctuations between the values of \(\beta\) among the oscillators in the network, i.e. \(\Delta \beta_i\). Indeed, as shown in Figure 3 for a network of 10 Repressilators, using the modified model provided by the algorithm stochastic synchronization of the repressilators is indeed observed to occur as expected.

Note that in (Garcia-Ojalvo et al. 2004), synchronization of a network of Repressilators is achieved even when the CFS generated by the algorithm are not satisfied. In that case, though, the effect of quorum sensing on the individual Repressilator dynamics is made stronger. Here, the algorithm gives CFS on the Repressilator parameters that can be used to achieve synchronization even in the presence of weaker quorum sensing. In order to make a comparison between the mathematical model used in (Garcia-Ojalvo et al. 2004) and the modified mathematical model derived by the algorithm, we use the order parameter, \(R\), also defined in (Garcia-Ojalvo et al. 2004) as
\[
R = \frac{\langle M^2 \rangle - \langle M \rangle^2}{\sum_{i=1}^{N} \langle b^2_i \rangle - \langle b_i \rangle^2},
\]
where \(M(t)\) denotes the average of \(b_i\) over the cell population, and \(\langle \bullet \rangle\) denotes the time average. Thus, by definition, \(R \approx 0\) if the network is not synchronized while \(R \approx 1\) if synchronization occurs. By using the same biochemical parameters for the quorum sensing mechanism we can compare the two mathematical models of the Repressilator circuit: the one used in (Garcia-Ojalvo et al. 2004) and the one we slightly modified according to the CFS generated by the algorithm described in this paper. We have found that our modified model ensures an increase of at least 10% in the steady-state value of the order parameter \(R\). We wish to emphasize that, to obtain a similar improvement of \(R\), in (Wang et al. 2006) the use of an external entrainment impulsive signal is suggested. By means of our algorithm, such an improvement is obtained by simply adjusting one of the Repressilator parameters rather than using any exogenous entrainment signal, according to the biological guidelines generated by the algorithm.

### 3 Discussion

Synchronization of biological clocks has been studied. It has been shown that an algorithm for the generation of Conditions for Synchronization (CFS) can be developed which makes use of recent results from dynamical systems theory. The algorithm consists of few key steps that can be used either to verify whether
an existing network synchronizes or to construct synthetic networks of oscillators that synchronize. Such conditions involve the network topology, the biochemical parameters of the oscillators in the network and their structure. In particular, the CFS were found to require a balance between the degradation and production rates of each species in the system, as explained in Section 3.3 of Supplementary Information.

The effectiveness of the algorithm proposed in this paper has been shown by using two representative examples aimed at expounding the viability of its use for both verification and design purposes. In the case of the Glycolysis process of yeast cells we found that our algorithm predicts the achievement of a stable synchronous regime. In the case of the Repressilator we found that is possible to modify the mathematical model proposed in (Garcia-Ojalvo et al. 2004) to achieve synchronization. We then compared the two models using an appropriately defined order parameter, typically used in the existing literature. We found a sensible improvement of the synchronization performance with an increase of the order parameter of at least 10%. Rather than using external entrainment signals as in (Wang et al. 2006), such an improvement was achieved by simply tuning one of the biological network parameters (the translation rate of one of the proteins) in order to fulfill the CFS generated by the algorithm. We wish to emphasize the viability and effectiveness of the proposed algorithm for the characterization of synchronization in biological networks. The algorithm furnish important guidelines that can be used to engineer synthetic biological networks of oscillators that synchronize and has therefore the potential to be used in a multitude of different applications where this is the primary objective.

4 Material and Methods

All simulations are performed in Matlab/SIMULINK v. 7.3.0 with variable step-size ODE solver ODE23s. Simulink models and simulation routines are available upon request.

5 Acknowledgment

We wish to thank Dr Diego di Bernardo and his group at the Telethon Institute of Genetics and Medicine (TIGEM) in Naples for all the helpful discussions and advice during the preparation of this manuscript. We thankfully acknowledge financial support from the European Union (6th Framework Program NEST PATHFINDER Project COBIOS no. 043379).

References


Table 1: Conditions for Synchronization (CFS).

<table>
<thead>
<tr>
<th>Statements for $J(i,j)$</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S1 - S2 - S3$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>$S1 - S2 - S3$</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
</tbody>
</table>

Table 2: Simplified Conditions for Synchronization (CFS).

<table>
<thead>
<tr>
<th>Statements for $J(i,j)$</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S1 - S2$ $\forall J(i,i)$ $S3$, $\forall J(i,j) \neq 0$</td>
<td>$\text{find } i :</td>
</tr>
<tr>
<td>$S1$ $true$ $\forall J(i,i)$</td>
<td>$\text{find positive } p_1, p_2, \ldots, p_m :</td>
</tr>
<tr>
<td>$S2$ $false$ $\forall J(i,i)$</td>
<td></td>
</tr>
<tr>
<td>$S3$ $false$ $\forall J(i,j) \neq 0$</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Simulation of system (10) using parameters given by our algorithm: the two nodes have different initial conditions (see the zoom in the left panel) but synchronization is achieved at steady state (see right panel)
Figure 2: Repressilator circuit and coupling mechanism

Figure 3: Stochastic synchronization for a network of nonidentical Repressilators: at steady state their trajectories are bounded due to the contraction property of their noise free virtual system