An Algorithm to Prove Contraction, Consensus, and Network Synchronization

Mario di Bernardo ∗ Giovanni Russo ∗∗ Jean-Jacques Slotine ∗∗∗

∗ University of Naples Federico II, (e-mail: mario.dibernardo@unina.it).
∗∗ University of Naples Federico II, (e-mail: giovanni.russo@unina.it)
∗∗∗ Massachusetts Institute of Technology, (e-mail: jjs@mit.edu)

Abstract: This paper introduces a novel algorithm to prove contraction of nonlinear dynamical systems. The algorithm is then used to devise novel strategies to achieve synchronization and consensus of networked control systems. All theoretical derivations are complemented by appropriate numerical simulations on a set of representative examples.

Keywords: Consensus, Networked Control Systems, Synchronization, Contraction Theory.

1. INTRODUCTION

Many complex systems in Nature and Technology can be described as networks of interacting agents communicating over the links of a graph describing their interconnections. Examples include neural systems, the world-wide-web, electrical power grids and ensembles of robotic vehicles (Newman (2003)). A key problem in this context is that of finding distributed algorithms for the coordination of such networks of dynamic agents (see e.g. Olfati-Saber and Murray (2004), Hui and Hadad (2008), Cortes and Bullo (2005), Chung and Slotine (2007)). Commonly, it is assumed that each agent has the ability of performing some computation and share information with its neighbors. The main advantages deriving from the use of such a distributed algorithm are modularity, fault-tolerance, integrability and extendibility.

Typical examples of coordination are consensus and synchronization of networks of dynamical systems where the objective is for all the agents in the network to reach asymptotically the same evolution. Different approaches have been proposed in the literature to solve this problem. Typically, these strategies are analyzed-designed by means of Lyapunov functions, as the coordination problem is reduced to a stability problem of some invariant set. However, in such questions we are typically interested in finding conditions guaranteeing the evolution of all the trajectories of the nodes of a network towards each other. For this reason, a possible approach to study agents cooperation is that of investigating the convergence properties of all solutions between each other rather than their convergence towards a specific solution or set, often known a priori.

A particularly powerful concept to assess convergence of solutions towards each other is nonlinear contraction analysis (Lohmiller and Slotine (1998)). Historically, ideas closely related to contraction can be traced back to Hartman (1961) and even to Lewis (1949) (see also Pavlov et al. (2004), Angeli (2002), and e.g. Lohmiller and Slotine (2005) for a more exhaustive list of related references).

As (Lohmiller and Slotine (1998)) points out, contraction is preserved through a large variety of systems combinations, and it represents in particular a natural tool for the design of nonlinear state observers. For autonomous systems and with constant metrics, the basic nonlinear contraction result above reduces to Krasovskii’s theorem (Slotine and Li (1990)) in the continuous-time case, and to the contraction mapping theorem in the discrete-time case (Lohmiller and Slotine (1998); Bertsekas and Tsitsiklis (1989)).

Applications to date of nonlinear contraction theory have been using formulations based on negative definite generalized Jacobians and Euclidian norms. As it was noticed in the original paper (Lohmiller and Slotine (1998)), other norms and their associated matrix measures (Vidyasagar (1993)) could be used to quantify contraction, leading to similar results but conditions in different algebraic forms. The aim of this paper is to explore in more detail this avenue showing that, as suggested in Russo and di Bernardo (2009c), it represents a natural approach for particular classes of systems, in particular those encountered in biochemistry and in network coordination problems, and develops systematic algorithmic tools for establishing contraction in these contexts.

Specifically, after presenting an overview of nonlinear contraction theory, we present a novel algorithm to prove or impose contraction for a nonlinear dynamical system. The algorithm is based on the use of matrix measures induced by non-Euclidean norms and can be used to prove contraction by looking at an appropriate graph constructed from the system generalized Jacobian. Using this graph, it is possible to derive a set of conditions that the system needs to satisfy in order for contraction to be guaranteed. Such conditions can then be used both to analyze a system of interest or design control inputs (or tune parameters) so that the system becomes contracting. Also, applying these conditions to virtual contracting systems as introduced in Wang and Slotine (2005) and Pham and Slotine (2007) leads to algorithms for consensus and synchronization.
problems in networks of dynamical systems. As representative examples, we take the general problem of constructing a self-synchronizing network of biological circuits and that of achieving consensus in networks of integrators via linear and nonlinear communication protocols. The theoretical predictions are validated by numerical simulations. We wish to emphasize that the approach presented in this paper is quite general and can be effectively extended to solve other problems such as flocking and synchronization in complex networks of dynamical systems.

2. NONLINEAR CONTRACTION THEORY

2.1 Basic result

The basic theorem of nonlinear contraction analysis (Lohmiller and Slotine (1998)) can be stated as

**Theorem 1. (Contraction).** Consider the deterministic system

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

(1)

where \( f \) is a smooth nonlinear function. If there exist a uniformly invertible matrix \( \Theta(x, t) \) such that the associated generalized Jacobian

\[ F = \left( \dot{\Theta} + \Theta \frac{\partial f}{\partial x} \right) \Theta^{-1} \]  

(2)

is uniformly negative definite, then all system trajectories converge exponentially to a single trajectory, with convergence rate \( |\lambda_{\text{max}}| \), where \( \lambda_{\text{max}} \) is the largest eigenvalue of the symmetric part of \( F \). The system is said to be contracting.

It can be shown conversely that the existence of a uniformly positive definite metric \( \Theta^T \Theta \) with respect to which the system is contracting is also a necessary condition for global exponential convergence of trajectories.

In the linear time-invariant case, a system is globally contracting if and only if it is strictly stable, with \( F \) simply being a normal Jordan form of the system and \( \Theta \) the coordinate transformation to that form.

2.2 Partial Contraction and Virtual Systems

A simple yet powerful extension to nonlinear contraction theory is the concept of partial contraction, which was introduced in Wang and Slotine (2005).

**Theorem 2. (Partial contraction).** Consider a nonlinear system of the form

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

and assume that the auxiliary system

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

is contracting with respect to \( y \). If a particular solution of the auxiliary \( y \)-system verifies a smooth specific property, then all trajectories of the original \( x \)-system verify this property exponentially. The original system is said to be partially contracting.

Indeed, the virtual, observer-like \( y \)-system has two particular solutions, namely \( y(t) = x(t) \) for all \( t \geq 0 \) and the solution with the specific property. Since all trajectories of the \( y \)-system converge exponentially to a single trajectory, this implies that \( x(t) \) verifies the specific property exponentially.

2.3 Convergence to a flow-invariant subspace

The results of Wang and Slotine (2005) are systematically extended in Pham and Slotine (2007) to global exponential convergence to flow-invariant linear subspaces, allowing in particular multiple groups of synchronized elements to coexist (so called poly-dynamics or, in the case of oscillators, poly-rhythms). A simplified version of the main result can be stated as follows (see Pham and Slotine (2007) for a proof).

**Theorem 3.** Consider a constant linear subspace \( \mathcal{M} \), flow-invariant under the system dynamics (1), and defined by \( Vx = 0 \) where the rows of the constant matrix \( V \) are orthonormal. If the reduced-order dynamics

\[ \dot{\hat{y}} = Vf(V^Ty, t) \]

is contracting with a constant metric, then all system trajectories converge exponentially to \( \mathcal{M} \).

Effects of noise on contracting systems and synchronization are studied in Pham et al. (2009).

2.4 Using Different Matrix Measures

It is now possible to consider a generalization (Lohmiller and Slotine (1998)) of Theorem 1 to different matrix measures, namely, for a square matrix \( A \),

- with \( |x|_2 = \left( \sum_{j=1}^{m} |x_j|^2 \right)^{\frac{1}{2}} \) (Euclidian norm), the induced matrix measure \( \mu_2 \) is the usual

\[ \mu_2(A) = \max_j \left( \frac{\lambda_j(A + A^*)}{2} \right) ; \]  

(3)

- with \( |x|_1 = \sum_{j=1}^{m} |x_j| \), the induced matrix measure, \( \mu_1 \), is

\[ \mu_1(A) = \max_j \left( a_{jj} + \sum_{i \neq j} |a_{ij}| \right) ; \]  

(4)

- with \( |x|_{\infty} = \max_{1 \leq j \leq m} |x_j| \), the induced matrix measure, \( \mu_{\infty} \), is

\[ \mu_{\infty}(A) = \max_j \left( a_{jj} + \sum_{i \neq j} |a_{ij}| \right) ; \]  

(5)

We can state

**Theorem 4.** (Contraction using different matrix measures). Consider the deterministic system (1). If there exists a uniformly invertible matrix \( \Theta(x, t) \) such that one of the above matrix measures of the generalized Jacobian \( F \) in (2) is uniformly strictly negative, then all system trajectories converge exponentially to a single trajectory, with a guaranteed convergence rate \( |\lambda_{\text{max}}| \) equal to the absolute value of the (negative) upper bound on the matrix measure of \( F \).

The remainder of the paper will be devoted to contraction results using the last two of these measures.

In the case that \( \Theta \) is constant, the generalized Jacobian (2) is simply \( F = \Theta \frac{\partial f}{\partial x} \Theta^{-1} \) and its measure corresponds to the matrix measure \( \mu_\Theta \) induced by the \( \Theta \)-norm,

\[ \mu(F) = \mu(\Theta \frac{\partial f}{\partial x} \Theta^{-1}) = \mu_\Theta(\partial f/\partial x) \]

This is the usual induced matrix measure.
3. AN ALGORITHM FOR PROVING CONTRACTION

We now propose iterative procedures for proving or imposing contraction for a nonlinear dynamical system, with a constant metric.

3.1 Algorithm outline

Consider again the nonlinear dynamical system (1): its differentiation with respect to the state variables yields the (state/time dependent) Jacobian matrix

\[ J := \begin{bmatrix} J(1,1) & J(1,2) & \cdots & J(1,m) \\ J(2,1) & J(2,2) & \cdots & J(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ J(m,1) & \cdots & J(m,m) \end{bmatrix}. \quad (6) \]

The outcome of the algorithm is to provide a set of conditions on the elements of \( J \) (and hence on the dynamics of \( f(x,t) \)) that can be used to prove contraction. Notice that (1) can represent, for instance, a closed loop control system, in which the control input is a function of the system state. Thus, the procedure presented here can be used both for checking if a system is contracting and for designing some control input guaranteeing contractivity (and hence some desired behavior).

The algorithm is based on the construction of a directed graph from the above system Jacobian. In particular, we first derive an adjacency matrix from \( J \), say \( \mathcal{A} \), using the following rules:

1. \( \forall i = 1, \ldots, m, \mathcal{A}(i,i) = 0; \)
2. if \( J(i,j) \neq 0 \), then \( \mathcal{A}(i,j) = \mathcal{A}(j,i) = 1; \)

Say \( \mathcal{G}_{\text{un}}(\mathcal{A}) \) the graph represented by \( \mathcal{A} \). We then associate to \( \mathcal{G}_{\text{un}}(\mathcal{A}) \) a rule providing the directions to its edges in order to obtain the directed graph, \( \mathcal{G}_d(\mathcal{A}) \). Define \( n_{0i} \) as the number of zero elements on the \( i \)-th row of \( J \) (or equivalently of \( \mathcal{A} \)) and

\[ \alpha(i,j) = \frac{\mid J(i,j) \mid}{\mid J(i,i) \mid} (m - n_{0i} - 1). \quad (7) \]

Then:
- if \( \alpha(i,j) \geq 1 \), the edge between node \( i \) and node \( j \) is directed from \( i \) to \( j \);
- if \( \alpha(i,j) < 1 \), the edge between node \( i \) and node \( j \) is directed from \( j \) to \( i \).

Given this graph, the main steps of the algorithm to prove contraction are:

1. check the uniform negativity of all the diagonal elements \( J(i,i) \);
2. construct the graph \( \mathcal{G}_d(\mathcal{A}) \) from \( J \) as described above and check if \( \mathcal{G}_d(\mathcal{A}) \) contains any loop.

If all diagonal elements are uniformly negative and \( \mathcal{G}_d(\mathcal{A}) \) does not contain any loop, then the system is contracting.

When the above conditions are not satisfied, the algorithm can be used to impose contraction for the system of interest by:

1. using, if possible, a control input to impose uniform negativity on all those diagonal elements \( J(i,i) \) that do not fulfill this condition;
2. re-direct (using an appropriate control input, or tuning system parameters) some edges of the graph \( \mathcal{G}_d(\mathcal{A}) \) in order to satisfy the loopless condition;
3. associate to each reverted edge (e.g. the edge between node \( i \) and node \( j \)) one of the following inequalities:

\[ \alpha(i,j) > 1, \text{ if the edge is reverted from } i \text{ to } j; \]
\[ \alpha(i,j) < 1, \text{ if the edge is reverted from } j \text{ to } i; \]

The outcome of the algorithm is then a set of conditions that can be used as guidelines to design an appropriate control input or tune the system parameters so that contraction for the system of interest is guaranteed.

3.2 Proof of the Algorithm

The procedure outlined in the previous Section is based on the following result.

**Theorem 5.** A dynamical system

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

is contracting if its Jacobian matrix, say \( J \), is such that

1. \( \forall i = 1, \ldots, m, J(i,i) < 0; \)
2. the graph \( \mathcal{G}_d(\mathcal{A}) \) as defined in the Appendix does not contain (directed) loops.

**Proof.** The idea of the proof is to derive a set of inequalities ensuring contraction of (1). These inequalities are then translated into some topological condition on the graph \( \mathcal{G}_d(\mathcal{A}) \). The first step of the proof is then to differentiate (1), yielding a state/time dependent Jacobian matrix of the form (6). From the discussion outlined in the above Section, to prove contraction we have to prove that there exists a negative matrix measure for \( J \). Namely, we use \( \mu_{\infty}(\Theta J \Theta^{-1}) \), where \( \Theta \) is a constant positive-definite diagonal matrix, i.e.

\[ \Theta := \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_m \end{bmatrix}. \]

Clearly, such a measure is negative if and only if:

1. \( \forall i = 1, \ldots, m, J(i,i) < 0; \)
2. \( \forall i, j = 1, \ldots, m, \exists p \in \mathbb{P}_1, \ldots, p_m : \)

\[ \frac{1}{p_i} \sum_{j=1, j \neq i}^m p_j |J(i,j)| < |J(i,i)|. \]

Notice that the first set of inequalities is satisfied from the hypotheses. Hence, to complete the proof, we have to show that the set of inequalities (2) corresponds to the loopless condition for \( \mathcal{G}_d(\mathcal{A}) \). Specifically, if we indicate with \( n_{0i} \) the number of zero elements on the \( i \)-th row of (6), then such a set is fulfilled if

\[ |J(i,i)| > \frac{p_i}{p_k} |J(i,j)| (m - n_{0i} - 1), \quad (8) \]

is satisfied \( \forall i, j = 1, \ldots, m, \text{with } i \neq j \). Inequalities (8) can be written in terms of the positive real numbers \( p_1, \ldots, p_m \):

\[ p_i > \alpha(i,j) p_j, \quad (9) \]

\( \forall i, j = 1, \ldots, m, i \neq j, \text{ where } \alpha(i,j) \text{ is defined as in (7)}. \)

It is clear that if the set of inequalities represented by (9) is consistent, i.e. there exists some set of values for \( p_1, \ldots, p_m \) satisfying such inequalities, then contraction will be immediately proven. We have then to show that
the loopless condition on \( G_d(A) \) is equivalent to the consistency of (9). We start by constructing a graph in which a node is associated to each parameter \( p_1, \ldots, p_m \in (9) \) and labeled as 1, \ldots, \( m \). The topology of the graph is then determined by the coefficients \( \alpha(i,j) \), given in (7).

In particular, node \( i \) and node \( j \) are linked if and only if \( \alpha(i,j) \neq 0 \). The direction of the edge linking \( i \) and \( j \) are instead assigned by the magnitude of \( \alpha(i,j) \):

- the edge is from \( i \) to \( j \) if \( \alpha(i,j) \geq 1 \);
- the edge is from \( j \) to \( i \) if \( \alpha(i,j) < 1 \);

Notice that the presence of loops in the graph implies the impossibility of finding \( p_1, \ldots, p_m \) satisfying (9). Indeed, suppose that one direct loop exists for such a graph, e.g. \( i \rightarrow j \rightarrow k \rightarrow i \). By construction, the inequalities involving \( p_i, p_j, p_k \) will then imply: \( p_i > p_j > p_k > p_i \). The proof is then concluded by noticing that the above graph corresponds to \( G_d(A) \).

### 3.3 Remarks

- Notice that the procedure presented above is based on the use of \( \mu_\infty(\Theta J \Theta^{-1}) \) for proving contraction. Clearly other matrix measures can also be used. It is easy to prove that, using \( \mu_1(\Theta J \Theta^{-1}) \), yields the same algorithmic procedure applied on \( J^T \). If this is the case, the resulting algorithm will follow the same logical steps as that of Section 3.1, with the only difference being the expression of \( \alpha(i,j) \):

\[
\alpha(i,j) := \frac{|J(j,i)| (m - c_{0i} - 1)}{|J(i,i)|}, \quad (10)
\]

where \( c_{0i} \) denotes the number of zero elements in the \( i \)-th column of \( J \). Clearly, one could also apply the algorithms after having performed some coordinate transformation.

- The outlined procedure can be easily implemented via a software interacting with the end user.

- The procedure presented here has got a clear physical interpretation if the system state variables are all homogeneous quantities. This is the case, for example, of molecular systems, i.e. systems composed by genes and proteins, where the state variables represent the concentrations of the species involved into the system. In this case, each term \( \alpha(i,j) \) represent a normalized production rate between species \( i \) and species \( j \) and the resulting set of inequalities provided by the algorithm points towards a balance of flows in the system (see e.g. Russo and di Bernardo (2009c) and Russo and di Bernardo (2009a)).

### 3.4 Example

As a representative example, we study a general externally-driven transcriptional module presented in Del Vecchio et al. (2008) which is currently under investigation in Russo et al. (2009). The module is described by the nonlinear dynamical system

\[
\begin{align*}
\dot{x} &= u(t) - \delta x + k_1 y - k_2 (E_T - y) x \\
\dot{y} &= -k_1 y + k_2 (E_T - y) x.
\end{align*}
\]

where \( x \) and \( y \) are the concentrations of two chemical species, \( E_T \) is the concentration of some enzyme which is supposed to be constant. Note that the term \((E_T - y)\) is a concentration and therefore must be non-negative. The parameters \( k_1 \) and \( k_2 \) are positive constants.

We will show, using the algorithm presented in Sec. 3, that this system is contracting (Russo et al. (2009)). As proved in Lohmiller and Slotine (1998), this in turn implies that (11) tends globally exponentially to a periodic solution of the same period as \( u(t) \).

Differentiation of (11) is then needed, yielding:

\[
J := \begin{bmatrix}
-\delta - k_2 (E_T - y) & k_1 + k_2 x \\
k_2 (E_T - y) & -k_1 - k_2 x
\end{bmatrix}.
\]

Clearly, in this case, the graph \( G_d(A) \) associated to \( J \) contains only two nodes, labeled as 1 and 2. Thus, the only possible loop in such a graph is that on length 1. To avoid the presence of such a loop, we have to ensure that the direction determined by \( \alpha(1,2) \) is the same as that determined by \( \alpha(2,1) \).

Computation of those two quantities in accordance with (10) is then needed. In particular:

\[
\begin{align*}
\alpha(1,2) &= \frac{k_2 (E_T - y)}{\delta + k_2 (E_T - y)} < 1, \\
\alpha(2,1) &= \frac{k_1 + k_2 x}{k_1 + k_2 x} = 1.
\end{align*}
\]

This, in turn implies that, following the schematic of Section 3.1, the directions determined by \( \alpha(1,2) \) and \( \alpha(2,1) \) are the same. In particular, the unique edge of the graph is directed from node 2 to node 1. Contraction is then proved, and therefore (11) is globally entrained by a periodic input.

We will now use the algorithm presented in the previous section to solve some representative synchronization and consensus problems.

### 4. SYNCHRONIZATION OF BIOLOGICAL NETWORKS

The problem that we address in this Section is that of tuning the parameters of synthetic biological oscillators so that, when coupled, they self synchronize.

The Repressilator is a synthetic biological circuit of three genes inhibiting each other in a cyclic way (Elowitz and Leibler (2000)). As shown in Figure 1, 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 C1 (\( B_i \)) translated from \( c_i \) inhibits expression of \( lacI \), completing the cycle. In García-Ojalvo et al. (2004), a modular addition to the classical Repressilator circuit is proposed with the aim of coupling different oscillators using the quorum sensing mechanism.

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 for the network is
Fig. 1. Repressilator circuit and coupling mechanism

\[ \begin{aligned}
\dot{a}_i &= -a_i + \frac{\alpha}{1 + C_i^2}, \\
\dot{b}_i &= -b_i + \frac{\alpha}{1 + A_i^2}, \\
\dot{c}_i &= -c_i + \frac{\alpha}{1 + B_i^2} + kS_i, \\
\dot{A}_i &= \beta_A a_i - d_A A_i, \\
\dot{B}_i &= \beta_B b_i - d_B B_i, \\
\dot{C}_i &= \beta_C c_i - d_C C_i, \\
\dot{S}_i &= -k_{s0} S_i + k_{s1} A_i - \eta (S_i - S_e), \\
\dot{S}_e &= -k_{se} S_e + \eta \eta_{ext} \sum_{j=1}^{N} (S_j - S_e)
\end{aligned} \]  (13)

having chosen the Hill coefficient equal to 2 as in Garcia-Ojalvo et al. (2004). We first assume that the Repressilator circuits on which the algorithm is applied are all identical. Referring to Garcia-Ojalvo et al. (2004) we set the biochemical parameters of the circuits, since no control is desired. Indeed, by direct inspection it is easy to check that 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 dynamics of the \(i\)-th Repressilator circuit is obtained. In this way the virtual system embeds the trajectories of all the network oscillators. Differentiation of (14) yields the Jacobian matrix

\[ J = \begin{bmatrix}
-1 & 0 & 0 & 0 & f_1 (C) & 0 & 0 \\
0 & -1 & 0 & f_1 (A) & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & f_1 (B) & 0 & f_2 (S) \\
\beta & 0 & 0 & -\beta & 0 & 0 & 0 \\
0 & \beta & 0 & 0 & -\beta & 0 & 0 \\
0 & 0 & \beta & 0 & 0 & -\beta & 0 \\
0 & 0 & 0 & k_{s1} & 0 & 0 & -k_{s0} - \eta \\
0 & 0 & 0 & 0 & 0 & 0 & k_{se} - k_{di,ff}
\end{bmatrix} \]  (15)

The diagonal elements of \( \tilde{J} \) are all negative, thus following the schematic of Section 3, \( \mathcal{G}_d (A) \) has to be constructed. In so doing, matrix \( \mathcal{A} \) is derived:

\[ \mathcal{A} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}. \]  (17)

From (17), \( \mathcal{G}_{\text{un}} (A) \) is obtained as shown in Figure 2 (left panel).

Fig. 2. Graphs associated to \( J \): \( \mathcal{G}_{\text{un}} (A) \) (left panel); \( \mathcal{G}_d (A) \) with state dependent edges (central panel); choice for \( \mathcal{G}_d (A) \) (right panel)

Then, computation of coefficients \( \alpha (i, j) \) (reported in Table 1) provides the directions of the edges of \( \mathcal{G}_{\text{un}} (A) \). Notice that the elements of the left column of Table 1 are all state dependent. This implies that the directions of the corresponding edges in \( \mathcal{G}_d (A) \) can be time-varying as they are associated to conditions which are functions of the state. Moreover, due to biochemical constraints (Garcia-Ojalvo et al. (2004)), \( \alpha (7, 4) < 1 \). However, the other coefficients in the table (right column) can be easily tuned since they depend only on biochemical parameters of the network (recall that in this case we can tune only the biochemical parameters of the circuits, since no control input is available). In Figure 2 (central panel), a partially directed graph is shown, obtained by assigning directions

\[
\begin{array}{c|c|c|c}
\alpha (i, j) & \text{Algebraic expression} & \alpha (i, j) & \text{Algebraic expression} \\
\hline
\alpha (1, 6) & \frac{2 \alpha}{(1 + C_i^2)} & \alpha (4, 1) & \frac{2 \alpha}{(1 + C_i^2)} \\
\alpha (2, 4) & \frac{2 \alpha}{(1 + A_i^2)} & \alpha (5, 2) & \frac{2 \alpha}{(1 + A_i^2)} \\
\alpha (3, 5) & \frac{2 \alpha}{(1 + B_i^2)} & \alpha (6, 3) & \frac{2 \alpha}{(1 + B_i^2)} \\
\alpha (3, 7) & \frac{2 \alpha}{(1 + C_i^2)} & \alpha (7, 4) & \frac{2 \alpha}{(1 + C_i^2)} \\
\end{array}
\]

where \( f_1 \) and \( f_2 \) denote the partial derivatives of decreasing and increasing Hill functions with respect to state variable of interest and \( k_{di,ff} = \eta \eta_{ext} / N \). Notice that matrix \( J \) has the form of a hierarchy (Lohmiller and Slotine (1998)). Furthermore, notice that \( J (8, 8) \) (associated to the quorum sensing dynamics) is negative definite. This implies that the algorithm can be applied to the submatrix \( J \) obtained by neglecting the last row and column of \( J \) and corresponding to the Repressilator circuit, i.e.
We can then state the following theorem.

**Theorem 6.** The communication protocol (18) guarantees that consensus is achieved if
\[
\frac{\partial \Phi}{\partial x} > 0.
\]

**Proof.** The proof is based on the use of an appropriate virtual system (Wang and Slotine (2005) and Pham and Slotine (2007)). In particular, for the network of interest a suitable virtual system is:
\[
\dot{x} = \Phi \left( \sum_{j \in N_x} \left( g(x_j) - g(x) \right) \right),
\]
where \( x \) is the virtual state variable and \( x_j \) is the set of state variables of the neighbors of \( x \), i.e. \( N_x \). Now, as described in Sec. 2, if such a system is contracting with respect to the virtual state variable, then all nodes will converge towards each other. Differentiation of the virtual system yields the (scalar) system Jacobian
\[
J := -N_x \frac{\partial \Phi}{\partial x} \frac{\partial g}{\partial x}.
\]

The theorem is then proved.

Note that, despite solving the consensus problem, the previous result does not give any a priori knowledge of the agreement point. Basically, this is due to the fact that the communication protocol \( u_i \) is assumed to be completely generic. The following result provides a simple sufficient condition for designing communication protocols solving the average consensus problem.

**Corollary 7.** Consider a network of integrators with \( u_i \) set as in (18). Then, the average consensus problem, i.e. \( \lim_{t \to \infty} x_i = \bar{x} = \sum_{i=1}^{N} x_i(0) \), is solved by any communication protocol satisfying the hypotheses of Theorem 6 and such that
\[
\sum_{j \in N} u_{j} = 0.
\]

**Proof.** The convergence analysis is exactly the same as that given in Theorem 6. Thus, all network trajectories converge towards each other. We have than to prove that an equilibrium point exists for the network: this, indeed, would imply that all trajectories converge to it (due to contraction). Clearly, since \( 0 = \sum_{j \in N} u_{j} \), the quantity \( \text{Ave}(x(t)) \) is invariant for the network dynamics. This immediately implies that all network trajectories converge to \( \bar{x} = \text{Ave}(x(0)) = \sum_{i=1}^{N} x_i(0) \).

**Remarks**

- As shown in Pham and Slotine (2007), any communication protocol of the form
  \[
  u_i = \sum_{j \in N_i} \left( g(x_j) - g(x_i) \right),
  \]
with \( g \) being a strictly increasing function, satisfies all the hypotheses of the above cited results. Thus, classical linear strategies typically used in the literature, i.e.
  \[
  u_i = \sum_{j \in N_i} a_{ij} (x_j - x_i),
  \]
are a particular case of Theorem 6 and Corollary 7.
• The strategies described above can be easily extended to encompass the case of \( n \)-th order integrators or nonlinear systems.

• In solving the consensus problem for a network of integrators, we did not make explicit use of the algorithm as the virtual system used to prove contraction has a very simple structure. It is worth emphasizing that in more general cases, such as higher-order linear or nonlinear dynamics at the nodes and/or more sophisticated coordination problems, the algorithm provides an invaluable tool for control system design.

5.2 Communication protocols design

The above theorem provides mild hypotheses on the distributed control strategy that ensures the desired behavior for the network. Thus, the proposed methodology gives some flexibility that can be used to satisfy some other requirements on the performance of consensus, e.g. selecting a desired agreement point, minimizing the control effort etc.

To illustrate this point, we make a comparison between the linear distributed control strategy (20) and a nonlinear protocol derived from Theorem 6 and Corollary 7. The comparison will be in terms of the maximal control effort, defined here as the maximum value reached by the communication protocol, \( u_i \).

The maximal control effort is indeed an important parameter for network design. Clearly, both communication protocols in (19) and (20) decrease as the disagreement between nearby agents is reduced. However, in practical applications, the control input is bounded by the capacity of the actuators. It is then important to ensure that the control effort does not surpass such limiting threshold.

Clearly, for protocol (20), we have
\[
|u_i| \leq \sum_{j \in N_i} |x_j(0)| + |x_i(0)|. 
\]

Thus, the maximum control effort is not predictable a priori since it depends solely on the initial values of the state variables.

We now show how, using the proposed methodology, it is possible to construct a nonlinear consensus protocol ensuring that the control effort remains lower than some desired threshold.

For example, suppose that we want the communication protocol \( u_i \) not to exceed the limiting value \( u_{\text{max},i} \). In accordance with Theorem 6 and Corollary 7, we need to choose a bounded function \( g \), which has also to be strictly increasing. A simple choice for such a function could be to set \( g(x) = \arctan(x) \).

Such a function is indeed strictly increasing and bounded by \(-\pi/2\) and \(\pi/2\). We can then consider the following nonlinear protocol
\[
u_i = \frac{u_{\text{max},i}}{\pi N_i} \sum_{j \in N_i} (\arctan(x_j) - \arctan(x_i)),
\]

which obviously guarantees \( |u_i| < u_{\text{max},i} \) while satisfying all the hypotheses needed for the virtual system to be contracting.

5.3 Numerical validations

In this section we provide numerical validation for the decentralized control strategies derived in Section 5.2. The simulations are performed using Matlab with ODE solver ode23s, having maximum step size equat to \(1e-9\). We compared the communication strategies (20) and (19) by using a random network of 100 agents having fixed topology, in which the initial states of the agents were chosen from a gaussian distribution of mean 0 and standard deviation 50. For the control strategy (19) we set \( u_{\text{max},i} = u_{\text{max}} = 10000 \). Figure 4 clearly shows that both the control strategies solve the average consensus problem (notice that they converge to the same agreement value). However, the nonlinear control strategy (19) ensures a faster convergence to the agreement point than the linear control strategy (20). Furthermore, it is interesting to note that while the maximum of the linear control action is greater than \( 10^4 \), the maximum of the nonlinear control strategy is considerably smaller with \( u_{\text{max}} \), being approximately \( 5 \times 10^3 \), as shown in Figure 5.
6. CONCLUSIONS

We presented a novel algorithm to prove or impose contraction for nonlinear dynamical systems. The key idea, when compared to previous approaches, is to use matrix measures induced by non-Euclidean norms. The resulting set of algebraic conditions was then turned into an algorithmic procedure to check contraction for a system of interest. At the core of the algorithm is the construction of a directed graph from the generalized Jacobian which is then required to be loopless in order for the system to be contracting. The possible implications of the algorithms for control system design were discussed and shown to be particularly effective on a set of representative examples.

After illustrating the algorithm on a transcriptional module, we considered the problem of constructing a self-synchronizing network of biological circuits. We showed that the algorithm gives a set of conditions that can be used to design the network so that its synchronization is guaranteed. We then considered the problem of achieving consensus in networks of integrators. After stating the problem, linear and nonlinear communication protocols were constructed using contraction theory. A nonlinear communication protocol was then proposed to satisfy possible bounds on the maximal control effort. Numerical simulations were used to validate the theoretical predictions.

We wish to emphasize that the novel algorithm presented in this paper can be used to a wide set of distributed and network control problems as it represents a flexible tool to analyse and design contracting dynamical systems.

REFERENCES


