



HAL
open science

Synchronization in Networks of Nonlinear Systems: Contraction Analysis via Riemannian Metrics and Deep-Learning for Feedback Estimation

Mattia Giaccagli, Samuele Zoboli, Daniele Astolfi, Vincent Andrieu, Giacomo Casadei

► **To cite this version:**

Mattia Giaccagli, Samuele Zoboli, Daniele Astolfi, Vincent Andrieu, Giacomo Casadei. Synchronization in Networks of Nonlinear Systems: Contraction Analysis via Riemannian Metrics and Deep-Learning for Feedback Estimation. 2023. hal-03801100v2

HAL Id: hal-03801100

<https://hal.science/hal-03801100v2>

Preprint submitted on 5 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Synchronization in Networks of Nonlinear Systems: Contraction Analysis via Riemannian Metrics and Deep-Learning for Feedback Estimation

Mattia Giaccagli, Samuele Zoboli, Daniele Astolfi, Vincent Andrieu, and Giacomo Casadei

Abstract— We consider the problem of exponential synchronization of a network of identical input-affine nonlinear time-varying systems connected through an undirected graph, in the presence of a leader. We tackle the problem with incremental stability tools. We propose sufficient metric-based conditions to design a distributed diffusive coupling feedback law in two frameworks. First, we consider full-state and (static) output feedbacks, where synchronization is obtained for every initial condition. Then, we show that synchronization can still be achieved in a smaller domain of attraction under milder assumptions. To balance the analytical difficulties of computing the proposed controller, we develop an algorithm based on deep neural networks (DNNs) for practical implementation.

Index Terms— Synchronization, contraction, multi-agent systems, incremental stability, deep learning, deep neural network.

I. INTRODUCTION

The problem of a group of agents trying to achieve an agreement is generally known as synchronization or consensus. In this work, we consider the problem of synchronization via distributed control feedback (i.e. diffusive coupling) of homogeneous networks (i.e. networks where the agents dynamics are identical). For linear systems, fundamental results were obtained in [1], [2]. See also [3, Section 5]. For nonlinear systems, most results exploit existing techniques developed for single-agent systems, specifically adapted to deal with a distributed framework. Among them, we recall passivity-based [4], dissipativity-based [5], and ISS approaches [6]. High-gain techniques, inherited from high-gain observers theory or high-gain domination approaches (see, e.g., [7], [8]), form another notable class of solutions. Finally, another very popular approach to solve the synchronization problem consists in exploiting tools derived from contraction and incremental stability theory (see [9]–[12]). Based on this framework, most of the results considered quadratic Lyapunov functions, or equivalently [12], Euclidean metrics. See, for instance, [13]–[19]. Only few investigated the use of nonlinear metrics, e.g. [20], [21].

In this work, we investigate the problem of global exponential synchronization of homogeneous networks, in which each agent is described by a nonlinear time-varying input-affine multi-input multi-output (MIMO) ordinary differential equations (ODEs). We look for the existence of a nonlinear diffusive coupling, namely a static

M. Giaccagli is with School of Electrical Engineering, Tel Aviv University, Ramat Aviv 69978, Israel (mattia@tauex.tau.ac.il)

S. Zoboli, D. Astolfi and V. Andrieu are with Université Claude Bernard Lyon 1, CNRS, LAGEPP UMR 5007, 43 boulevard du 11 novembre 1918, F-69100, Villeurbanne, France (name.surname@univ-lyon1.fr).

G. Casadei is with Laboratoire Ampere Dpt. EEA of the Ecole Centrale de Lyon, Université de Lyon, 69134 Ecully, France (giacomo.casadei@ec-lyon.fr).

The research leading to these results is partially funded by ANR via project DELICIO, number ANR-18-CE40-0010, and partially found by the Israel Science Foundation grant ISF 3621/21

distributed state/output control feedback. We consider the case of undirected and leader-connected graphs, that is, there exists an agent (the leader) who can send information to the other nodes without receiving any. In the rest of the network, the communication links are bi-directional. Following a contraction-based approach, we investigate the use of nonlinear Riemannian metrics (see, e.g., [11], [22]). We propose a solution to the synchronization problem based on the existence of a solution to a static partial differential inequality (PDI) which conceptually extends the stabilizability Riccati-like algebraic inequality (see, e.g., [2, Section II.C]). This allows to consider classes of systems with a nonlinear input gain. However, the proposed PDI is very complex to verify analytically. Therefore, to deal constructively with such a complexity, we show that synchronization can be achieved in a regional context under less stringent assumptions (i.e. without the so-called “Killing assumption” and the “integrability condition”, see below). Thanks to this relaxation, we provide a formulation of a practical algorithm based on Deep Neural Networks (DNNs) to check the solvability of such a PDI. Note that the proposed approach presents conceptual similarities with the control contraction metrics [23]. Therein, however, the control action is obtained by path-integrating along the geodesic (see [23, equation (6)]) by solving an optimization problem.

Note that the use of DNNs for solving PDIs is not new to the Machine Learning community, see, e.g., [24], [25]. In this context, we recall the recent results on the use of Deep Learning tools for the estimation of a Riemannian metric [26]–[29]. To show the potential of this approach, we consider the problem of synchronizing a network of Lorentz oscillators in which the input gain is highly nonlinear (and for which existing techniques cannot be easily applied). We highlight that this work is an extension of the authors’ conference work [22], where preliminary results on the synchronization of leader-connected undirected networks of SISO time-invariant nonlinear systems were proposed.

Notation: \mathbb{N} , resp. \mathbb{R} , is the set of natural, resp. real, numbers, $\mathbb{R}_{\geq 0} := [0, +\infty)$, $\mathbb{R}_{> 0} := (0, +\infty)$. We denote Given N matrices $A_i \in \mathbb{R}^{n_i \times n}$ for $i = 1, \dots, N$, we denote with $\text{col}(A_1, \dots, A_N)$ the matrix $(A_1^\top, \dots, A_N^\top)^\top \in \mathbb{R}^{(n_1 + \dots + n_N) \times n}$ and with $\text{diag}(A_1, \dots, A_N)$ the square matrix with A_1, \dots, A_N on the main diagonal and zeros everywhere else. Given a vector $x \in \mathbb{R}^n$ and a set $S \subset \mathbb{R}^n$, we denote the Euclidean distance of x with respect to S as $|x|_S := \inf_{z \in S} |x - z|$. Given a 2-tensor $P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ taking symmetric values and a vector field $f : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ both C^1 , we denote the Lie derivative of the tensor P along f as $L_f P(x, t)$, defined as

$$L_f P(x, t) := P(x, t) \frac{\partial f}{\partial x}(x, t) + \frac{\partial f}{\partial x}^\top(x, t) P(x, t) + \mathfrak{d}_f P(x, t)$$

$$\mathfrak{d}_f P(x, t) := \lim_{h \rightarrow 0} \frac{P(X(x, t+h, t)) - P(x, t)}{h} + \frac{\partial P}{\partial t}(x, t),$$

with $X(x, t, t_0)$ being the solution to $\frac{\partial}{\partial t} X(x, t, t_0) = f(X(x, t, t_0), t)$, with $X(x, t, t) = x$, for all $t \geq t_0$ in time-

existence of solutions. Given a 2-tensor $P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ and a vector field $g : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ both C^1 (resp., a C^1 matrix function $g : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times m}$), we say that g is a *Killing vector field* (or that it satisfies the Killing vector property) with respect to P , if $L_g P(x, t) = 0$ (resp. $L_{g_i} P(x, t) = 0$ for all $i = 1, \dots, m$, with g_i denoting the i -th column of g) for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$. Note that the Killing vector property is trivially satisfied in case both P, g are constant matrices.

II. PRELIMINARIES AND PROBLEM STATEMENT

A. Graph theory

In a general framework, a communication graph is described by a triplet $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ in which $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ is a set of $N \subset \mathbb{N}$ *vertices* (or nodes), $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of *edges* e_{jk} that models the interconnection between the vertices with the flow of information from vertex j to vertex k *weighted* by the (k, j) -th entry $a_{kj} \geq 0$ of the *adjacency matrix* $\mathcal{A} \in \mathbb{R}^{N \times N}$. We denote by $\mathcal{L} \in \mathbb{R}^{N \times N}$ the *Laplacian matrix* of the graph, defined as $\ell_{kj} = -a_{kj}$ for $k \neq j$, and $\ell_{kj} = \sum_{i=1}^N a_{ki}$ for $k = j$, where $\ell_{j,k}$ is the (j, k) -th entry of \mathcal{L} . We denote with \mathcal{N}_i the set of in-neighbors of node i , i.e. the set $\mathcal{N}_i := \{j \in \{1, \dots, N\} \mid e_{ji} \in \mathcal{E}\}$. A time-invariant graph is said to be *weakly connected* if and only if \mathcal{L} has only one trivial eigenvalue $\lambda_1(\mathcal{L}) = 0$ and all other eigenvalues $\lambda_2(\mathcal{L}), \dots, \lambda_N(\mathcal{L}) \in \mathbb{C}$ have strictly positive real parts (see [3, Theorem 5.1]).

In this article, we will consider *leader-connected* undirected graphs. With leader-connected, we mean that we assume the existence of a *leader* (i.e. the graph contains at least one spanning tree with the leader as a root). The leader is labeled (without loss of generality) as node 1 and has a set of in-neighbors that is the empty set, i.e. $\mathcal{N}_1 = \emptyset$. In other words, no node in the network can send information to node 1. We also assume the network to be *undirected*, meaning that we assume communication links to be bi-directional (i.e. $e_{ij} = e_{ji}$ for every $i, j = 2, \dots, N$), except, of course, to the edges that nodes in the network share with the leader. By considering such a graph structure, the following property is proved in [30].

Lemma 1 *Suppose the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is undirected and leader-connected. Then the Laplacian \mathcal{L} can be partitioned as $\mathcal{L} = \begin{bmatrix} 0 & 0 \\ \mathcal{L}_{21} & \mathcal{L}_{22} \end{bmatrix}$. Moreover, there exists a strictly positive real number $\mu > 0$ such that $\mathcal{L}_{22} \succeq \mu I$*

B. Synchronization with nonlinear diffusive coupling

In this article we consider a network of N agents. As we labeled the leader as node 1, its dynamics is defined by

$$\dot{x}_1 = f(x_1, t), \quad y_1 = h(x_1, t), \quad (1a)$$

where $x_1 \in \mathbb{R}^n$ is the state of the leader and $y_1 \in \mathbb{R}^p$ its output, while the dynamics of the other $N - 1$ nodes in the network are described as

$$\begin{aligned} \dot{x}_i &= f(x_i, t) + g(x_i, t)u_i, \\ y_i &= h(x_i, t), \end{aligned} \quad i = 2, \dots, N. \quad (1b)$$

where $x_i \in \mathbb{R}^n$ is the state of node i , $u_i \in \mathbb{R}^m$ is the control action on node i and $y_i \in \mathbb{R}^p$ is its output. We suppose that f, g, h are C^2 functions in their first argument and piecewise continuous in the second. We denote the state of the entire network as

$$\mathbf{x} := \text{col}(x_1, \dots, x_N) \in \mathbb{R}^{Nn}. \quad (2)$$

Furthermore, we denote with $\mathcal{X}_i(x_i^\circ, t, t_0)$ the trajectory of agent i evaluated at time $t \geq t_0$ such that $\mathcal{X}_i(t_0) = x_i^\circ$, and with

$\mathcal{X}(\mathbf{x}^\circ, t, t_0)$ the trajectory of the entire network (2) evaluated at initial condition $\mathbf{x}^\circ \in \mathbb{R}^{Nn}$, initial time $t_0 \in \mathbb{R}$ at time $t \geq t_0$. Our synchronization objective is to design a nonlinear diffusive coupling, namely a distributed feedback control law of the form

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} \left[\varphi(x_i, y_j, t) - \varphi(x_i, y_i, t) \right] = - \sum_{j=1}^N \ell_{ij} \varphi(x_i, y_j, t) \quad (3)$$

for all $i = 2, \dots, N$, for some C^1 function $\varphi : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}^m$, that stabilizes the dynamics (1) on the so-called *leader-synchronization manifold* \mathcal{D} defined as

$$\mathcal{D} := \{\mathbf{x} \in \mathbb{R}^{Nn} \mid x_i = x_1, \text{ for all } i \in \{1, \dots, N\}\}, \quad (4)$$

where the states of all the agents of the network agree with the leader. By construction, the i -th agent uses only the output information y_j of its neighborhoods $j \in \mathcal{N}_i$ and its own local information (y_i, x_i) . Furthermore, the control action u_i is equal to zero on the synchronization manifold. In other words, when consensus is achieved, no correction term is needed for each individual agent. As a consequence, stabilizing all the agents on a desired equilibrium point is generally not a valid solution in such a framework. We formalize our synchronization problem as follows.

Problem 1 (Leader synchronization) *The distributed feedback control law (3) solves the leader-synchronization problem for the network (1) if the manifold \mathcal{D} defined in (4) is globally uniformly exponentially stable for the closed-loop dynamics*

$$\begin{aligned} \dot{x}_1 &= f(x_1, t) \\ \dot{x}_i &= f(t, x_i) - g(t, x_i) \sum_{j \in \mathcal{N}_i} \ell_{ij} \varphi(x_i, y_j, t), \quad i = 2, \dots, N, \end{aligned} \quad (5)$$

namely, there exist positive constants k and $\lambda > 0$ such that for all (\mathbf{x}°, t_0) in $\mathbb{R}^{Nn} \times \mathbb{R}$ solutions of (5) are defined for all $t \geq t_0$ and

$$|\mathcal{X}(\mathbf{x}^\circ, t, t_0)|_{\mathcal{D}} \leq k \exp(-\lambda(t - t_0)) |\mathbf{x}^\circ|_{\mathcal{D}}, \quad \forall t \geq t_0. \quad (6)$$

In order to solve our leader synchronization problem, the following standing assumption is supposed to hold all along the paper.

Assumption 1 *The graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is undirected and leader-connected. Moreover, for each (x_1°, t_0) in $\mathbb{R}^n \times \mathbb{R}$ the trajectory of (1a) exists for all $t \geq t_0$.*

III. MAIN RESULTS

A. State-feedback design

In this section, we first consider the problem in which agents can exchange with their neighbors the full state information, namely $y_i = x_i$ in (1b). In compact form, we consider a network of N agents described by

$$\begin{aligned} \dot{x}_1 &= f(x_1, t) \\ \dot{x}_i &= f(x_i, t) + g(x_i, t)u_i, \quad i = 2, \dots, N. \end{aligned} \quad (7)$$

First, we suppose that the pair f, g satisfies a controllability assumption which is stated as follows.

Assumption 2 *There exist a C^1 matrix function $P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ taking symmetric positive definite values and positive real numbers $\underline{p}, \bar{p}, \rho, \lambda > 0$ such that the following holds for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$*

$$\begin{aligned} L_f P(x, t) - \rho P(x, t) g(x, t) g(x, t)^\top P(x, t) &\leq -2\lambda P(x, t), \\ \underline{p} I &\preceq P(x, t) \preceq \bar{p} I. \end{aligned} \quad (8)$$

Assumption 2 can be seen as a Riccati-like inequality (see the remark below), where P is a matrix function. This matrix function P can be employed to endow with a Riemannian metric \mathbb{R}^n . At each x in \mathbb{R}^n , we define the scalar product $\langle a, b \rangle_P = a^\top P(x)b$, (a, b) in \mathbb{R}^{2n} . The (uniform) upper and lower bound on P are required in order to guarantee that the induced norm is equivalent to the Euclidean one $|x| = \sqrt{x^\top x}$. These bounds are needed to show that, if the distance associated with the norm P between any node and the leader decreases along the solutions, so it does also with the Euclidean one.

Remark. For linear systems of the form $\dot{x} = Ax + Bu$, Assumption 2 boils down to the well-known algebraic Riccati inequality (ARI) $PA + A^\top P - \rho P B B^\top P \preceq -2\lambda P$ which admits a solution under the mild assumption that (A, B) is stabilizable. In this case, a stabilizing control action is given by $u = -\kappa B^\top P$ for any $\kappa \geq \rho$. Furthermore, we remark that such a design possesses the infinite-gain margin property [31]. Such a property will play a fundamental role in the following distributed analysis, as it will provide a synchronizing control law that is "robust" with respect to the graph topology, see [3, Section 5].

Assumption 3 *The matrix function g has the Killing vector field (see Notation) property with respect to P , namely*

$$L_g P(x, t) = 0, \quad \forall (x, t) \in \mathbb{R}^n \times \mathbb{R}. \quad (9)$$

Assumption 4 *The vector field Pg satisfies an integrability condition in the sense that, denoting $g = [g_1 \dots g_m]$, there exists a C^2 function $\alpha = (\alpha_1, \dots, \alpha_m)$, $\alpha_\iota : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}$ for $\iota = 1, \dots, m$, satisfying*

$$\frac{\partial \alpha_\iota}{\partial x}(x, t) = g_\iota(x, t)^\top P(x, t), \quad \forall (x, t) \in \mathbb{R}^n \times \mathbb{R}. \quad (10)$$

The Killing vector condition (9) in Assumption 3 implies that distances in the metric P between different trajectories of the dynamical equation $\dot{x} = g(x, t)$ are invariant. It means that signals entering in the direction of g may change the trajectories but do not change the distance between different trajectories. The integrability condition (10) in Assumption 4 is introduced since the synchronization analysis will be made through incremental properties and thus by analyzing the Jacobian of the vector fields. We remark that, for linear systems, both properties are always satisfied. Indeed P is Euclidean (and thus constant) and so it is $g(x, t) = B$. This implies that the Killing vector assumption (9) holds and the function α in (10) is $\alpha(x, t) = B^\top P x$.

Remark. Note that Assumption 3 is needed in order to obtain synchronization for every initial condition of the agents, i.e. to make the leader-synchronization manifold globally exponentially stable. As we will show in Section IV, such an assumption can be relaxed in exchange of obtaining asymptotic stability of \mathcal{D} with a domain of attraction included in a compact set.

As last remark, we highlight that Assumption 2 and Assumption 3 recover the design proposed in [23, Section III.A]. We're now ready to show the main result of this section.

Theorem 1 *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents (1) and let Assumptions 1 to 4 hold. Then, for any $\kappa \geq \frac{\rho}{2\mu}$, with μ given by Lemma 1, the distributed state-feedback control law (3) with*

$$\varphi(x_i, y_j, t) = \kappa \alpha(x_j, t), \quad (11)$$

and α satisfying (10), solves the synchronization Problem 1 for the network of agents given in (1).

Proof: The main goal is to show that the norm of the difference between any agent x_i and the leader x_1 exponentially decreases to zero. Therefore, let us consider the following change of coordinates

$$x_i \mapsto \tilde{x}_i := x_i - x_1, \quad i = 2, \dots, N$$

and let us collect all the vectors \tilde{x}_i as $\tilde{x} := \text{col}(\tilde{x}_2, \dots, \tilde{x}_N)$ and define $z = x_1$. Since $\ell_{ij} = 0$ for all $j \notin \mathcal{N}_i$, the dynamics of the error \tilde{x}_i for all $i = 2, \dots, N$ with the control law (3), (11) can be rewritten as

$$\begin{aligned} \dot{\tilde{x}}_i &= f(z + \tilde{x}_i, t) - f(z, t) \\ &\quad - \kappa g(z + \tilde{x}_i, t) \left[\sum_{j=2}^N \ell_{ij} \alpha(z + \tilde{x}_j, t) + \ell_{i1} \alpha(z, t) \right]. \end{aligned}$$

Note that there is no term on $g(z, t)$ since no control action is acting on the leader. Since $\sum_{j=1}^N \ell_{ij} = 0$ for all $i = 1, \dots, N$, we can add the term $\kappa g(z + \tilde{x}_i, t) \left(\sum_{j=1}^N \ell_{ij} \right) \alpha(z, t) = 0$ and get

$$\begin{aligned} \dot{\tilde{x}}_i &= f(z + \tilde{x}_i, t) - f(z, t) \\ &\quad - \kappa g(z + \tilde{x}_i, t) \sum_{j=2}^N \ell_{ij} [\alpha(z + \tilde{x}_j, t) - \alpha(z, t)]. \quad (12) \end{aligned}$$

Furthermore, note that in these new coordinates, the leader-synchronization manifold defined in (4) corresponds to the origin of the \tilde{x} -dynamics.

Now, given $(z^\circ, \tilde{x}^\circ, t_0)$ in $\mathbb{R}^{Nn} \times \mathbb{R}$, let $\mathcal{T} > t_0$ be the time of existence of the solution of (12) initialized from $(z^\circ, \tilde{x}^\circ)$ at time t_0 . For t in $[t_0, \mathcal{T}]$, let $(\mathcal{Z}(t), \tilde{\mathcal{X}}(t))$ denotes this solution. Consider the function $\Gamma : [0, 1] \times [t_0, \mathcal{T}] \mapsto \mathbb{R}^{Nn}$, with $\Gamma = (\Gamma_2, \dots, \Gamma_N)$ which satisfies $\Gamma(s, t_0) = s \tilde{x}^\circ$, and where $\Gamma_i, i = 2, \dots, N$, is the solution of the following ordinary differential equation for $t_0 \leq t < t_0 + \mathcal{T}$

$$\begin{aligned} \frac{\partial \Gamma_i}{\partial t}(s, t) &= f(\zeta_i(s, t), t) - f(\mathcal{Z}(t), t) \\ &\quad - \kappa g(\zeta_i(s, t), t) \sum_{j=2}^N \ell_{ij} (\alpha(\zeta_j(s, t), t) - \alpha(\mathcal{Z}(t), t)) \end{aligned}$$

where we denoted $\zeta_i(s, t) = \mathcal{Z}(t) + \Gamma_i(s, t)$. Note that, by uniqueness of the solution, Γ satisfies

$$\Gamma(0, t) = 0, \quad \Gamma(1, t) = \tilde{\mathcal{X}}(t), \quad \forall t \in [t_0, \mathcal{T}]. \quad (13)$$

Consider now the function V defined by

$$V = \sum_{i=2}^N V_i, \quad V_i(\cdot) = \int_0^1 \frac{\partial \Gamma_i}{\partial s}(s, \cdot)^\top P(\zeta_i(s, \cdot), \cdot) \frac{\partial \Gamma_i}{\partial s}(s, \cdot) ds. \quad (14)$$

Note that we have for all (k, l) in $\{1, \dots, n\}^2$ we have

$$\begin{aligned} &\frac{d}{dt} [P(\zeta_i(s, t), t)_{kl}] \\ &= \frac{\partial P_{kl}}{\partial x}(\zeta_i(s, t), t) \frac{\partial \zeta_i}{\partial t}(s, t) + \frac{\partial P_{kl}}{\partial t}(\zeta_i(s, t), t), \\ &= \frac{\partial P_{kl}}{\partial x}(\zeta_i(s, t), t) \left[f(\mathcal{Z}(t) + \frac{\partial \Gamma_i}{\partial t}(s, t)) \right] + \frac{\partial P_{kl}}{\partial t}(\zeta_i(s, t), t). \end{aligned}$$

This implies that for all vector ν in \mathbb{R}^n , and $i = 2, \dots, N$,

$$\begin{aligned} \frac{d}{dt} \left[\nu^\top P(\zeta_i(s, t), t) \nu \right] &= \nu^\top \mathfrak{d}_f P(\zeta_i(s, t), t) \nu \\ &\quad - \kappa \sum_{j=2}^N \left[\ell_{ij} \sum_{\iota=1}^m \nu^\top \mathfrak{d}_{g_\iota} P(\zeta_i(s, t), t) \nu \right. \\ &\quad \left. \times (\alpha_\iota(\zeta_j(s, t), t) - \alpha_\iota(\mathcal{Z}(t), t)) \right]. \end{aligned}$$

By using the Killing vector assumption (9) and the integrability one in (10), the time derivative of V_i becomes

$$\begin{aligned} \dot{V}_i(t) &= \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}(s, t)^\top L_f P(\zeta_i(s, t), t) \frac{\partial \Gamma_i}{\partial s}(s, t) \right. \\ &\quad - 2\kappa \frac{\partial \Gamma_i}{\partial s}(s, t)^\top \sum_{j=2}^N \ell_{ij} P(\zeta_i(s, t), t) g(\zeta_i(s, t), t) \\ &\quad \left. \times g(\zeta_j(s, t), t)^\top P(\zeta_j(s, t), t) \frac{\partial \Gamma_j}{\partial s}(s, t) \right] ds. \quad (15) \end{aligned}$$

With the following notations,

$$\begin{aligned} D(s, t) &:= \text{diag} \{ L_f P(\zeta_i(s, t), t) \}_{i=2, \dots, N}, \\ \Psi(s, t) &:= \text{col} \left\{ P(\zeta_i(s, t), t) g(\zeta_i(s, t), t) \frac{\partial \Gamma_i}{\partial s}(s, t) \right\}_{i=2, \dots, N}, \end{aligned}$$

we compute the derivative of V as follows

$$\begin{aligned} \dot{V}(t) &= \int_0^1 \left[\frac{\partial \Gamma^\top}{\partial s}(s, t) D(t) \frac{\partial \Gamma}{\partial s}(s, t) - 2\kappa \Psi(s, t) \mathcal{L}_{22} \Psi(s, t)^\top \right] ds \\ &\leq \int_0^1 \left[\frac{\partial \Gamma^\top}{\partial s}(s, t) D(t) \frac{\partial \Gamma}{\partial s}(s, t) - 2\kappa \mu \Psi(s, t) \Psi(s, t)^\top \right] ds \end{aligned}$$

where in the second step we used Assumption 1 and Lemma 1. Therefore, by selecting $\kappa \geq \frac{\rho}{2\mu}$ with ρ satisfying inequality (8) and $\mu > 0$ given by Lemma 1, we get $\dot{V}(t) \leq -\lambda V(t)$. From Gronwall's Lemma, this implies $V(t) \leq \exp(-\lambda(t - t_0)) V(t_0)$ for all $t \in [t_0, t_0 + \mathcal{T})$. Moreover, with (13) it yields

$$V_i(t) \geq \underline{p} \int_0^1 \frac{\partial \Gamma_i}{\partial s}(s, t)^\top \frac{\partial \Gamma_i}{\partial s}(s, t) ds \geq \underline{p} \|\tilde{\mathcal{X}}_i(t)\|^2. \quad (16)$$

Moreover, since

$$V(t_0) \leq \bar{p} \|\tilde{x}^\circ\|^2, \quad (17)$$

it yields for all t in $[t_0, t_0 + \mathcal{T})$ that

$$\|\tilde{\mathcal{X}}(t)\|^2 \leq \exp(-\lambda(t - t_0)) \frac{\bar{p}}{\underline{p}} \|\tilde{x}^\circ\|^2.$$

Hence, remembering the definition $\tilde{x}_i = x_i - z$ and the fact that the leader trajectory is well-defined for all positive times, this implies that the trajectories are complete in positive time (i.e. $\mathcal{T} = +\infty$). Moreover, by equivalence of norms in finite dimensional spaces it follows that there exist two strictly positive real numbers $\underline{c}, \bar{c} > 0$ such that $\underline{c} |\mathcal{X}(x^\circ, t, t_0)|_{\mathcal{D}} \leq |\tilde{\mathcal{X}}(t)| \leq \bar{c} |\mathcal{X}(x^\circ, t, t_0)|_{\mathcal{D}}$, for all $t \geq t_0$, which implies (6) and concludes the proof. ■

B. Output-feedback designs

We consider in this section the case of output feedback distributed control laws, see, e.g. [8, Section 2]. To this end, we particularize our result to the case of a network of agents described as

$$\begin{aligned} \dot{x}_1 &= f(x_1, t), & y_1 &= h(x_1, t), \\ \dot{x}_i &= f(x_i, t) + u_i, & y_i &= h(x_i, t), \end{aligned} \quad (18)$$

for all $i = 2, \dots, N$, with $y_i \in \mathbb{R}^p$ and $u_i \in \mathbb{R}^n$. The following result establishes a set of sufficient conditions for the existence of a distributed output feedback nonlinear diffuse coupling of the form (11). Assumptions 2 and 3 are then modified as follows.

Assumption 5 *There exist a C^1 matrix function $P : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n \times n}$ taking symmetric positive definite values, and real numbers $\underline{p}, \bar{p}, \lambda, \rho > 0$ such that the following holds for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$*

$$L_f P(x, t) - \rho \frac{\partial h}{\partial x}(x, t)^\top \frac{\partial h}{\partial x}(x, t) \preceq -2\lambda P(x, t), \quad (19)$$

$$pI \preceq P(x, t) \preceq \bar{p}I.$$

Assumption 6 *The vector field $\beta : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ defined as*

$$\beta(x, t) = P(x, t)^{-1} \frac{\partial h}{\partial x}(x, t)^\top$$

is a Killing vector for P , i.e. $L_\beta P(x, t) = 0$ for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$.

Note that, in the linear case, condition (19) corresponds to the detectability condition $PA + A^\top - \rho C^\top C \preceq -2\lambda P$, provided a control gain of the form $\varphi(x_i, y_i, t) = -P^{-1} C^\top y_i$. Assumptions 5 and 6 coincide with those in [32] used in the context of observer design with Riemannian metrics. Finally, we have the following result.

Proposition 1 *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents (18) and suppose Assumptions 1, 5 and 6 hold. Then, for any $\kappa \geq \frac{\rho}{2\mu}$, with μ given by Lemma 1, the distributed output-feedback control law (3) with $\varphi(x_i, y_j, t) = \kappa \beta(x_i, t) y_j$ solves the synchronization Problem 1 for the network (18).*

Proof: The proof follows similar arguments as the one in Theorem 1 and it is omitted for space reasons. A proof for the time-invariant single-output case can be found in [22, Theorem 2]. ■

We remark that, differently from the previous case, in the output feedback context the i -th agent needs the knowledge of its own state x_i in order to implement the proposed control law. In other words, the local information does not coincide with the distributed one.

The previous result can be also easily adapted to the context of a Riemannian incremental passivity condition for a network of systems of the form

$$\begin{aligned} \dot{x}_1 &= f(x_1, t), & y_1 &= h(x_1, t), \\ \dot{x}_i &= f(x_i, t) + g(x_i, t) u_i, & y_i &= h(x_i, t), \end{aligned} \quad (20)$$

for all $i = 2, \dots, N$ and $y_i, u_i \in \mathbb{R}^m$, namely $m = p$. In this case, such a result is obtained under the solution of the Riccati-like condition in Assumption 5. The Assumption 6 is however modified as follows.

Assumption 7 *The following condition holds for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$*

$$P(x, t) g(x, t) = \frac{\partial h}{\partial x}(x, t)^\top, \quad L_g P(x, t) = 0. \quad (21)$$

Note that Assumption 5 together with 7 extend to the Riemannian case the Euclidean passivity conditions given in [19]. In the linear case, we recover $PA + A^\top - \rho C^\top C \preceq -2\lambda P$, with $PB = C^\top$. Finally, we have then the following result. The proof is omitted for space reasons.

Proposition 2 *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents (20) and suppose Assumptions 1, 5 and 7 hold. Then, for any $\kappa \geq \frac{\rho}{2\mu}$, with μ given by Lemma 1, the distributed output-feedback control law (3) with $\varphi(x_i, y_j, t) = \kappa y_j$ solves the synchronization Problem 1 for the network (20).*

IV. RELAXING THE INTEGRABILITY AND KILLING CONDITIONS

The main limitation of the approach presented in Section III is the complexity of finding a metric P solving (8) and, at the same time, satisfying the Killing vector field property in (9) and the integrability condition in (10). In this section, we aim to provide a practical solution to such limitations. We show that synchronization can still be obtained under an approximate integrability condition or an approximate Killing vector assumption. Note that the type of result is different in the two contexts. Indeed, global results can still be achieved in case of an approximation of the integrability condition, provided the control gain is not selected too large. However, only semi-global results can be obtained when relaxing the Killing vector assumption. For space reasons, in the rest of this section we consider the state-feedback case developed in Section III-A. Nonetheless, similar conclusions can be drawn for the output feedback designs presented in Section III-B.

A. About the integrability condition

Instead of Assumption 4, consider the following one.

Assumption 4' *There exist a C^2 function $\alpha : \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^m$ and a scalar $\varepsilon > 0$ such that, for $\iota = 1, \dots, m$, the following holds*

$$\left\| \frac{\partial \alpha_\iota}{\partial x}(x, t) - g_\iota(x, t)^\top P(x, t) \right\| \leq \varepsilon, \quad \forall (t, x) \in \mathbb{R} \times \mathbb{R}^n. \quad (22)$$

Theorem 2 *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents (1) and let Assumptions 1, 2, 3 and 4' hold. Moreover, assume there exists a positive real number \bar{g} such that $|g_\iota(x, t)| \leq \bar{g}$ for all (x, t) in $\mathbb{R}^n \times \mathbb{R}$ and $\iota = 1, \dots, m$. Let μ be given by Lemma 1 and let $\bar{L} = \max_{i,j} |\ell_{ij}|$ where (ℓ_{ij}) is the Laplacian matrix associated to the graph. Then, if ε in Assumption 4' is selected as $\varepsilon \in [0, \varepsilon^*]$ with $\varepsilon^* = \frac{\lambda_{\mu P}}{\rho N \bar{L} m \bar{p} \bar{g}}$, there exists κ^* such that for any $\kappa \in [\frac{\rho}{2\mu}, \kappa^*]$, the distributed state-feedback control law (3)-(11) solves the synchronization Problem 1.*

Proof: The proof is identical to the proof of Theorem 1 up to equation (15). Recalling the compact notation $\zeta_i(s, t) = \mathcal{Z}(t) + \Gamma_i(s, t)$, with Assumption 2, for each i in $\{2, \dots, N\}$, the function V_i defined in (14) satisfies for all t in $[t_0, T)$

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}^\top D \frac{\partial \Gamma_i}{\partial s} - 2\kappa \mu \Psi \Psi^\top + T_i \right] ds$$

where we used the compact notation introduced in the proof of Theorem 1, we omitted the arguments for space reasons, and

$$\begin{aligned} T_i(s, t) &= -2\kappa \frac{\partial \Gamma_i}{\partial s}(s, t)^\top \sum_{j=1}^N \ell_{ij} \sum_{\iota=1}^m P(\zeta_i(s, t), t) g_\iota(\zeta_i(s, t), t) \\ &\times \left(\frac{\partial \alpha_\iota}{\partial x}(\zeta_j(s, t), t) - g_\iota(\zeta_j(s, t), t)^\top P(\zeta_j(s, t), t) \right) \frac{\partial \Gamma_j}{\partial s}(s, t). \end{aligned}$$

With Assumption 4', the bound on P , g and the ℓ_{ij} 's, the term T_i can be bound as follows

$$|T_i(t, s)| \leq 2\kappa \left\| \frac{\partial \Gamma_i}{\partial s}(s, t, t_0) \right\| N \bar{L} m \bar{p} \bar{g} \varepsilon \left\| \frac{\partial \Gamma_j}{\partial s}(s, t) \right\|$$

where $\bar{L} = \max_{i,j} \{|\ell_{ij}|\}$. Consequently, with Assumption 1, Lemma 1, and following the proof of Theorem 1, it yields,

$$\begin{aligned} \dot{V}(t) &\leq \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s, t)^\top D(s, t) \frac{\partial \Gamma^\top}{\partial s}(s, t) - 2\kappa \mu \Psi(s, t) \Psi^\top(s, t) \right. \\ &\quad \left. + 2\kappa N \bar{L} m \bar{p} \bar{g} \varepsilon \left\| \frac{\partial \Gamma_i}{\partial s}(s, t) \right\| \left\| \frac{\partial \Gamma_j}{\partial s}(s, t) \right\| \right] ds. \end{aligned}$$

Using the fact that $2\|v\|\|w\| \leq v^\top v + w^\top w$ for any two vectors v and w in \mathbb{R}^n , this gives,

$$\dot{V}(t) \leq \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s, t)^\top \text{diag}\{\Upsilon_i\}_{i=2, \dots, N} \frac{\partial \Gamma}{\partial s}(s, t) \right] ds,$$

where $\Upsilon_i(\zeta, t) = L_f P + 2\kappa \bar{c} P - 2\kappa \mu P g^\top g P$, with the functions P and g depending on (ζ, t) and $\bar{c} = \frac{N \bar{L} m \bar{p} \bar{g} \varepsilon}{\rho}$. With Assumption 2, this implies $\Upsilon_i(\zeta, t) \leq (2\kappa \bar{c} - \lambda)P + (\rho - 2\kappa \mu)P g^\top g P$. Note that with the choice of ε in the statement of the theorem, we have $\frac{\rho}{2\mu} < \frac{\lambda}{2\bar{c}}$. Consequently, for each $\kappa \in (\frac{\rho}{2\mu}, \frac{\lambda}{2\bar{c}})$ it implies $\dot{V}(t) \leq -\tilde{\lambda} V(t)$ for all $t \geq t_0$, where $\tilde{\lambda} = 2\kappa \bar{c} - \lambda$ is a positive real number. The proof completes following the lines of the proof of Theorem 1. ■

B. About the Killing vector field

When the equality constraint in Assumption 3 is replaced by an approximation, the global synchronization may be lost. However, it is shown in the following theorem that provided $|L_g P|$ is small enough, a semi-global result can be obtained.

Theorem 3 *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of systems (1). Suppose Assumption 1, 2, and 4 hold. Let $\kappa \geq \frac{\rho}{\mu}$ be fixed. Assume $|g(x)| \leq \bar{g}$ for all x in \mathbb{R}^n . Then, for each $\bar{x} > 0$ there exist $k, \varepsilon > 0$ such that, if the following holds*

$$|L_{g_\iota} P(x, t)| \leq \varepsilon, \quad \forall (x, t, \iota) \in \mathbb{R}^n \times [t_0, \infty) \times \{1, \dots, m\}, \quad (23)$$

then, for all (\mathbf{x}°, t_0) in $\mathbb{R}^{Nn} \times \mathbb{R}$ such that $\|\mathbf{x}^\circ\|_{\mathcal{D}} \leq \bar{x}$, the solution of (5) with the distributed state-feedback control law (3), (11) is defined for all $t \geq t_0$ and

$$\|\mathcal{X}(\mathbf{x}^\circ, t, t_0)\|_{\mathcal{D}} \leq k \exp(-\frac{\lambda}{3}(t - t_0)) \|\mathbf{x}^\circ\|_{\mathcal{D}}, \quad \forall t \geq t_0. \quad (24)$$

Proof: Let $\bar{x}, t_0 > 0$ and let $\mathbf{x}^\circ \in \mathbb{R}^{Nn}$ satisfy $\|\mathbf{x}^\circ\|_{\mathcal{D}} \leq \bar{x}$. Assume that (23) is satisfied for some positive real number ε that will be selected later on. As in the proof of Theorem 1, consider the function V defined in (14). With (16) and (17), there exists two positive real numbers $(\underline{c}_V, \bar{c}_V)$ such that

$$\underline{c}_V \|\mathcal{X}(\mathbf{x}^\circ, t)\|_{\mathcal{D}}^2 \leq V(t) \leq \bar{c}_V \|\mathcal{X}(\mathbf{x}^\circ, t)\|_{\mathcal{D}}^2.$$

With Assumption 2, and 4, for each i in $\{2, \dots, N\}$, the function V_i defined in (14) satisfies for all t in $[t_0, T)$

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}^\top D \frac{\partial \Gamma_i}{\partial s} - 2\kappa \mu \Psi \Psi^\top + \tilde{T}_i \right] ds$$

where we used the compact notation introduced in the proof of Theorem 1, we omitted the arguments for space reasons, and

$$\tilde{T}_i(t, s) = -\kappa \frac{\partial \Gamma_i}{\partial s}(s, t)^\top \sum_{\iota=1}^m L_{g_\iota} P(\zeta_i(s, t), t) \sum_{j=2}^N \ell_{ij} \tilde{\alpha}_\iota \frac{\partial \Gamma_i}{\partial s}(s, t).$$

with $\tilde{\alpha}_\iota(\zeta_j) := \alpha_\iota(\zeta_j(s, t), t) - \alpha_\iota(\mathcal{Z}(t), t)$. Note that, with Assumption 4 we have

$$\begin{aligned} \tilde{\alpha}_\iota(\zeta_j(s, t)) &= \int_0^1 \frac{\partial \alpha_\iota}{\partial x}(\mathcal{Z}(t) + r \Gamma_j(s, t), t) \Gamma_j(s, t) dr \\ &= \int_0^1 \frac{\partial \alpha_\iota}{\partial x}(\mathcal{Z}(t) + r \Gamma_j(s, t), t) \int_0^s \frac{\partial \Gamma_j}{\partial s}(\nu, t) d\nu dr. \end{aligned}$$

Hence,

$$|\tilde{\alpha}_\iota(\zeta_j)| \leq \bar{p} \bar{g} \int_0^1 \left| \frac{\partial \Gamma_j}{\partial s}(\nu, t) \right| d\nu \leq \bar{p} \bar{g} \frac{1}{2} \int_0^1 1 + \left| \frac{\partial \Gamma_j}{\partial s}(\nu, t) \right|^2 d\nu.$$

Employing that $\int_0^1 \|\frac{\partial \Gamma_i}{\partial s}(s, t)\|^2 ds \leq \frac{V_i(t)}{p}$, and with (23), this gives

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma^\top}{\partial s} D \frac{\partial \Gamma}{\partial s} - 2\kappa\mu\Psi\Psi^\top + V_i(t)\varepsilon\kappa\bar{c} \left(1 + \frac{V_j(t)}{p} \right) \right] ds$$

where $\bar{c} = mN\bar{L}\frac{\bar{p}}{p}\bar{g}$ where $\bar{L} = \max_{ij} |\ell_{ij}|$. Consequently, following the proof of Theorem 1, we obtain, with $\kappa \geq \frac{p}{2\mu}$,

$$\dot{V} \leq -(\lambda - \bar{c}\kappa\varepsilon)V + \frac{\bar{c}\kappa\varepsilon}{p} \sum_{i=2}^N V_i^2 \leq -\left(\lambda - \bar{c}\kappa\varepsilon - \frac{\bar{c}\kappa\varepsilon}{p}V\right)V.$$

Finally, selecting $\varepsilon < \max\left\{\frac{\lambda}{3\bar{c}\kappa}, \frac{\lambda p}{3\bar{c}\kappa\bar{c}Vx}\right\}$ it implies $\dot{V} \leq -\frac{\lambda}{3}V$ for all $t \geq t_0$, concluding the proof. ■

V. DEEP LEARNING FOR METRIC ESTIMATION

As mentioned in the previous section, a drawback of the proposed approach lies in the fact that metrics may not be easy to find in the Riemannian scenario. Moreover, even when a metric has been found, designing a control law satisfying the integrability condition (10) may not be straightforward. One way to overcome such difficulties is to rely on Machine Learning tools to obtain approximate solutions, thus leveraging on Theorem 3 and Theorem 2. In what follows, we combine the proposed control design with Deep Learning tools. In recent years, Deep Neural Networks (DNNs) turned out to be effective tools for solving complex differential equations, see, e.g., [24], [25]. As a matter of fact, multiple works began mixing learning tools and control. Such a combined framework tackles the complexity of computing control theoretic exact solutions by exploiting deep approximators. Hence, the idea is to set up and approximately solve an optimization problem aimed at circumventing the need for an analytic metric. Once a suitable metric has been found via a first DNN, we train a second one to satisfy the integrability condition.

A. Related results

Similar approaches already appeared in the literature (see e.g., [26]–[28], [33]). In [26], the authors propose a convex optimization problem to compute a suitable metric. Yet, they successively suggest approximating the solution via a DNN. Hence, the convex optimization is solved on a finite number of samples and the DNN provides a continuous interpolation through those points. This overcomes the need of solving such an optimization in each point of the state space. In [28] the authors propose learning Control Contraction Metrics (CCMs) [23] to solve output tracking problems. However, when considering the approximation error induced by the learning procedure, the results offer probabilistic convergence guarantees to trajectories close to the reference one. Moreover, the learned controller needs to be structured in order to always verify the assumptions. Finally, [27] proposes a Siamese DNN structure [34]. Similarly to our scenario, both [27] and [28] aim at minimizing a loss function defined by the matrix conditions required for contraction. Once such a function reaches 0, the DNN provides the entries of a suitable metric for each point in the training/test datasets.

Even if related, our solution differs on some fundamental points. With respect to [28], we do not impose constraints on the controller structure. Hence, we do not affect the expressivity of the selected approximator. In addition, our approach is not based on CCMs. Hence, while the proposed loss functions are similar in the sense they involve sign definiteness-related costs, their components are different. Moreover, we also optimize the components parameters, which results in more plastic constraints. On a further note, our sign definite cost relies on eigenvalues. While possibly resulting in more complex conditions, this choice ensures sign definiteness of the

matrix. This differs from the random sampling approach in [28]. As a final remark, due to the choice of the loss components, our solution offers asymptotic convergence guarantees even when assumptions are only approximately satisfied, see Theorem 2 and Theorem 3. With respect to [27], we rely on the continuous-time framework. Hence, we avoid the need of a Siamese network by computing the DNN Jacobian. Note that, usually, such a Jacobian can be easily obtained thanks to the automatic differentiation tools provided by common libraries such as Pytorch [35]. Second, as previously stated, we add a separate estimator which looks for the best parameters in the cost function. It works jointly with the DNN during the optimization process. Finally, we rely on (10) instead of computing the control law via approximate integration over the geodesic. This greatly simplifies the algorithm, since geodesics are not easy to find in general.

B. Proposed approach

We now describe the proposed algorithm. Let us consider the problem of finding a suitable approximation of the metric first. The neural metric is constructed as

$$P(x, \theta') = \begin{bmatrix} P_1(x, \theta') & \cdots & P_n(x, \theta') \\ \vdots & \ddots & \vdots \\ P_n(x, \theta') & \cdots & P_p(x, \theta') \end{bmatrix},$$

where $p = \frac{n(n+1)}{2}$ is the total number of entries to be learned, the vector $\varrho = (P_1(x, \theta'), \dots, P_p(x, \theta'))^\top$ is the output of the neural network $\text{DNN}_P : \mathbb{R}^n \times \mathbb{R}^{n\theta'} \mapsto \mathbb{R}^p$ and $\theta' \in \mathbb{R}^{n\theta'}$ is the vector of DNN_P parameters. To train the DNN_P parameters, we rely on Theorem 2 to relax the existence of a primitive for $g(x)^\top P(x, \theta')$ and on Theorem 3 to loosen the constraint posed by the Killing vector field property (9). We set up an optimization problem asking for the minimization of the following cost function

$$J_P(x, \theta') = \sum_{i=1}^4 w_i J_i(x, \theta'), \quad (25)$$

being $w = (w_1, \dots, w_4)$ a vector of (positive) scalar weights and

$$J_i(x, \theta') = \ln(\max(\Re\{\lambda_M(M_i)\}, 0) + 1),$$

with $i = 1, \dots, 4$, λ_M being the maximum eigenvalue and M_i defined as

$$\begin{aligned} M_1 &= L_f P(x, \theta') - \rho P(x, \theta') g(x) g^\top(x) P(x, \theta') + \varepsilon I \\ M_2 &= L_g P(x, \theta') - \epsilon I, \quad M_3 = -L_g P(x, \theta') - \epsilon I, \\ M_4 &= -P(x, \theta') + pI \end{aligned}$$

where $\rho, \epsilon, p > 0$ are positive scalars with $\varepsilon > \epsilon$ and where $\Re\{\lambda\}$ is the real part of the complex number $\lambda \in \mathbb{C}$. Note that each cost J_i serves the purpose of satisfying a particular condition for the neural metric. While J_1 provides a positive cost if the contraction condition (8) is not satisfied, J_2 and J_3 encourage the boundedness of $L_g P$, thus relaxing the Killing vector condition (9), and J_4 steers the solution towards positive definite matrices, see (8). Note that the upperbound is always satisfied as we optimize our algorithm in a compact set X . The natural logarithm is used as a regularization term between costs J_i . It allows the rescaling of widely different costs to similar values and a more precise selection of their importance through the weight vector w . In parallel to DNN_P , we train a parameter estimator outputting the values of $\rho, \varepsilon, \epsilon, p$. The estimator and DNN_P work together, trying to minimize (25). Note that if the cost reaches 0, all the contraction conditions are satisfied for the dataset and the learned estimator outputs, hence learning can be stopped. The second step is to find a suitable law approximating the

integrability condition (10). We train the parameters $\theta'' \in \mathbb{R}^{n\theta''}$ of the second network $\text{DNN}_\alpha : \mathbb{R}^n \times \mathbb{R}^{n\theta'} \mapsto \mathbb{R}^m$ such that

$$J_\alpha(x, \theta'') = \left| \frac{\partial \text{DNN}_\alpha}{\partial x}(x, \theta'') - g(x)^\top P(x, \theta') \right|^2 \quad (26)$$

is minimized. The full learning procedure is summarized by Algorithm 1. Clearly, the DNNs can be trained only on a dataset \mathbb{D} of finite size. Yet, DNNs are typically Lipschitz-continuous functions. Hence, similarly to [27, Section IV], we provide a verification tool via the following proposition, to assess the satisfaction of contraction conditions over compact sets once the training is over.

Proposition 3 *Let $S \subset \mathbb{R}^n$ be an arbitrary compact set and $\mathbb{D} \subset \mathbb{R}^n$ a set with a finite number of elements and let $r > 0$ be such that*

$$S \subseteq \cup_{x_i \in \mathbb{D}} \mathcal{B}(x_i, r), \quad \mathcal{B}(x_i, r) := \{x \in \mathbb{R}^n : |x - x_i| < r\}.$$

Let $M : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$ be a Lipschitz-continuous matrix-valued function, with Lipschitz constant L_M , taking symmetric values and such that $M(x_i) \preceq -2qI$ for all $x_i \in \mathbb{D}$ and for some $q > 0$. If q, r, L_M are such that $q > rL_M$, then $M(x) \preceq -qI, \forall x \in S$.

Proof: By Lipschitz-continuity of M we have

$$\begin{aligned} M(x) &= M(x_i) + M(x) - M(x_i) \preceq M(x_i) + |M(x) - M(x_i)|I \\ &\preceq M(x_i) + L_M|x - x_i|I \preceq -q \left(2 - \frac{|x - x_i|}{r} \right) I \end{aligned}$$

for an arbitrary $x_i \in \mathbb{D}$. Then, $M(x) \preceq -qI$ for all $x \in \mathcal{B}(x_i, r)$. The result follows from the fact that $S \subseteq \cup_{x_i \in \mathbb{D}} \mathcal{B}(x_i, r)$. ■

Proposition 3 implies that if the dataset is composed of a sufficiently fine grid, then the learned properties extend to the points in between. Hence, we can obtain a valid metric over a compact set by learning on a finite number of samples. Similar reasoning can be proposed for the feedback law α .

Since the estimated metric is a DNN and $g \in C^2$, their product is Lipschitz continuous on a compact set. Since α is also modeled as a DNN, by selecting smooth activation functions its Jacobian is continuous. Following similar arguments to those used in Proposition 3, we can finally guarantee that a bounded error on a grid translates to a bounded error on a compact set including it.

VI. ILLUSTRATION

In the following, we apply the proposed algorithm to a leader-synchronization problem¹. We consider a network of $N = 6$ identical Lorenz attractors. Such systems are particularly interesting, since they can present a chaotic behavior. Each agent $i = 1, \dots, N$ is described by the following dynamics

$$\begin{cases} \dot{x}_{i,1} = a(x_{i,2} - x_{i,1}) + u_i \\ \dot{x}_{i,2} = x_{i,1}(b - x_{i,3}) - x_{i,2} + (2 + \sin(x_{i,1}))u_i \\ \dot{x}_{i,3} = x_{i,1}x_{i,2} - cx_{i,3} \end{cases}$$

with $x_i = (x_{i,1}, x_{i,2}, x_{i,3}) \in \mathbb{R}^3$ and where $a = 10, b = 28, c = \frac{8}{3}$, guaranteeing the chaotic behavior. We consider the control matrix $g(x) = (1, 2 + \sin(x_{i,1}), 0)$ to exclude the possibility of feedback linearizing solutions. The agents communicate with each other following the leader-connected graph represented in Figure 1a.

We code and train our fully-connected DNNs and estimator using PyTorch [35]. For the metric network, we select an architecture composed of 4 hidden layers, with size 30, 20, 20, 10 respectively

¹The code for reproducing the experiments proposed in this section can be found at <https://github.com/SamueleZoboli/Control-learning-multiagent-lorenz.git>

Algorithm 1 DNN-based controller learning

- 1: Input: Dataset of $(x, f(x), g(x), \frac{\partial f}{\partial x}(x), \frac{\partial g}{\partial x}(x))$, $\text{DNN}_P, \text{DNN}_\alpha$;
 - 2: **while** $J_P(x, \theta') \neq 0$ **do**
 - 3: Train DNN_P and the estimator with (25);
 - 4: **end while**
 - 5: Train the DNN_α with (26);
 - 6: Set the distributed law $u_i = -\kappa \sum_{j=1}^N \ell_{ij} \text{DNN}_\alpha(x_j, \theta'')$.
-

and tanh activation functions. The output layer passes through a saturation function as a final activation, limiting the single elements of the metric. The second network presents 3 hidden layers, with size 30, 20, 10 respectively and tanh activation functions. We select the identity function as output layer activation function.

We select a weight vector $w = (1, 10, 10, 20)$, directing the learning towards positive matrices first and successively satisfying the Killing-less assumptions and the contraction condition. We train both the networks and the estimator using Adam optimizer [36]. The learning rate for the metric network and the estimator is set as 3×10^{-3} , while DNN_α uses a learning rate of 5×10^{-3} . The DNNs learning rate are scheduled according to a cosine annealing policy [37], while the estimator one remains constant. We train the neural metric and the estimator over 100 epochs (yet stopped after 15 epochs due to the cost reaching 0) and the second DNN over 200 epochs. For both of the learning phases (the metric learning and the integrability learning), the dataset is composed by 2×10^5 samples coming from a Gaussian distribution $\mathcal{N}(0, 10)$. We use 80% of the dataset as the training set, with a batch size of 512. The remaining 20% is used as test set.

We select a $\kappa = 5$ and we apply the controller in a noisy-measurements scenario, i.e., $u_i = \varphi(x_i + \nu_i)$ where $\nu_i \sim \mathcal{N}(0, 0.2)$ represents some Gaussian measurement noise. This allows testing the robustness properties of the proposed neural control law. Each agents' initial condition is randomly sampled from a Gaussian distribution $\mathcal{N}(0, 20)$. Figures 1b and Figure 1c show the controller performances once the DNNs have been trained. Figure 1b presents the mean and standard deviation between agents of the norm of the error with respect to the leader trajectory. Figure 1c directly shows the state trajectories of each agent. As synchronization is achieved, we can see that the DNN optimized with (25) provides a suitable estimated metric, while the one trained with (26) effectively learns an approximate primitive of $g(x)^\top P(x, \theta')$. The parameter estimator provided a decay rate $\varepsilon \approx 4.7$ and $\rho \approx 36.3$. From Figure Figure 1c it is possible to see that the agents quickly synchronize, despite having significantly different initial conditions.

VII. CONCLUSIONS

We studied the problem of exponential synchronization of a homogeneous network of input-affine nonlinear time-varying systems connected through an undirected graph with the presence of a leader. We proposed a set of sufficient conditions based on Riemannian metrics to design a distributed diffusive coupling state and output feedback for every initial condition. Then, we showed that synchronization can still be achieved under milder assumptions in a smaller domain of attraction. To relax computational difficulties, we provided an algorithm based on DNNs for practical implementation, and tested it in a challenging environment. Future studies will try to generalize the proposed conditions to more general networks by using different Lyapunov functions and/or more complex analysis tools.

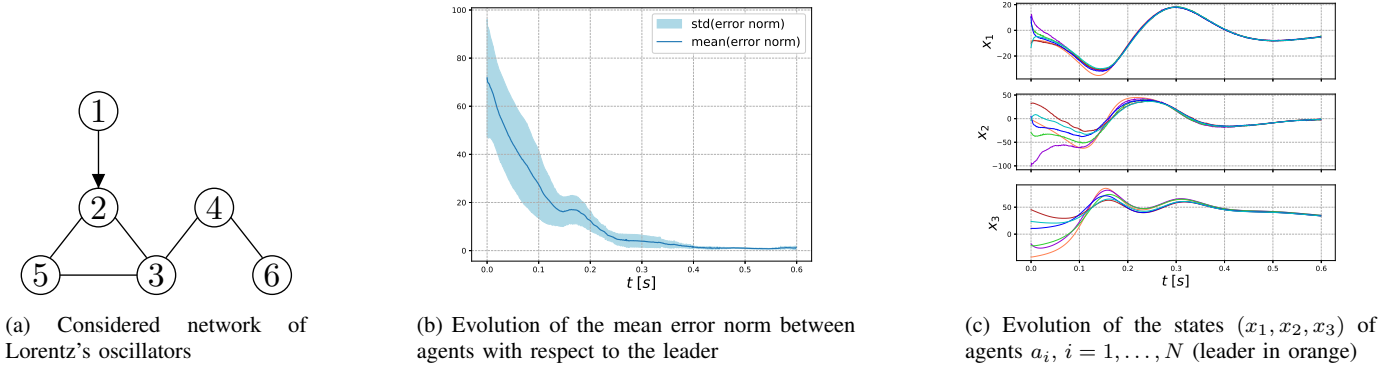


Fig. 1: Synchronization of Lorentz's oscillators

REFERENCES

- [1] L. Scardovi and R. Sepulchre, "Synchronization in networks of identical linear systems," in *47th IEEE Conference on Decision and Control*, 2008, pp. 546–551.
- [2] Z. Li, Z. Duan, G. Chen, and L. Huang, "Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 213–224, 2009.
- [3] A. Isidori, *Lectures in feedback design for multivariable systems*. Springer, 2017.
- [4] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.
- [5] G. Stan and R. Sepulchre, "Analysis of interconnected oscillators by dissipativity theory," *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 256–270, 2007.
- [6] G. Casadei, D. Astolfi, A. Alessandri, and L. Zaccarian, "Synchronization in networks of identical nonlinear systems via dynamic dead zones," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 667–672, 2019.
- [7] N. Chopra and M. Spong, "Output synchronization of nonlinear systems with relative degree one," in *Recent advances in learning and control*. Springer, 2008, pp. 51–64.
- [8] A. Isidori, L. Marconi, and G. Casadei, "Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory," *IEEE Transactions on Automatic Control*, vol. 59, no. 10, pp. 2680–2691, 2014.
- [9] F. Forni and R. Sepulchre, "A differential lyapunov framework for contraction analysis," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 614–628, 2013.
- [10] D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Transactions on Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.
- [11] V. Andrieu, B. Jayawardhana, and L. Praly, "Transverse exponential stability and applications," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3396–3411, 2016.
- [12] A. Davydov, S. Jafarpour, and F. Bullo, "Non-euclidean contraction theory for robust nonlinear stability," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6667–6681, 2022.
- [13] W. Wang and J.-J. E. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biological cybernetics*, vol. 92, no. 1, pp. 38–53, 2005.
- [14] M. Arcak, "Certifying spatially uniform behavior in reaction–diffusion pde and compartmental ode systems," *Automatica*, vol. 47, no. 6, pp. 1219–1229, 2011.
- [15] F. Zhang, H. Trentelman, and J. Scherpen, "Fully distributed robust synchronization of networked Lur'e systems with incremental nonlinearities," *Automatica*, vol. 50, no. 10, pp. 2515–2526, 2014.
- [16] M. Coraggio, P. DeLellis, S. J. Hogan, and M. Di Bernardo, "Synchronization of networks of piecewise-smooth systems," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 653–658, 2018.
- [17] V. Andrieu and S. Tarbouriech, "LMI conditions for contraction and synchronization," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 616–621, 2019.
- [18] H. Yin, B. Jayawardhana, and R. Reyes-Báez, "Pinning synchronization of heterogeneous multi-agent nonlinear systems via contraction analysis," *IEEE Control Systems Letters*, 2021.
- [19] A. Pavlov, E. Steur, and N. van de Wouw, "Nonlinear integral coupling for synchronization in networks of nonlinear systems," *Automatica*, vol. 140, p. 110202, 2022.
- [20] Z. Aminzare and E. Sontag, "Synchronization of diffusively-connected nonlinear systems: Results based on contractions with respect to general norms," *IEEE transactions on network science and engineering*, vol. 1, no. 2, pp. 91–106, 2014.
- [21] S. Jafarpour, P. Cisneros-Velarde, and F. Bullo, "Weak and semi-contraction for network systems and diffusively coupled oscillators," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1285–1300, 2022.
- [22] M. Giaccagli, V. Andrieu, D. Astolfi, and G. Casadei, "Sufficient metric conditions for synchronization of leader-connected homogeneous nonlinear multi-agent systems," *IFAC-PapersOnLine*, vol. 54, no. 14, pp. 412–417, 2021.
- [23] I. Manchester and J. Slotine, "Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.
- [24] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.
- [25] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. Karniadakis, "Physics-informed neural networks (pinns) for fluid mechanics: A review," *Acta Mechanica Sinica*, pp. 1–12, 2022.
- [26] H. Tsukamoto, S. Chung, and J. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021.
- [27] L. Wei, R. McCloy, and J. Bao, "Discrete-time contraction-based control of nonlinear systems with parametric uncertainties using neural networks," *Computers & Chemical Engineering*, vol. 166, p. 107962, 2022.
- [28] D. Sun, S. Jha, and C. Fan, "Learning certified control using contraction metric," in *Conference on Robot Learning*, 2021, pp. 1519–1539.
- [29] S. Zoboli, S. Janny, and M. Giaccagli, "Deep learning-based output tracking via regulation and contraction theory," *22nd IFAC World Congress*, 2022.
- [30] C. Godsil and G. Royle, *Algebraic graph theory*. Springer, 2001.
- [31] M. Giaccagli, V. Andrieu, S. Tarbouriech, and D. Astolfi, "Infinite gain margin, contraction and optimality: an LMI-based design," *European Journal of Control*, p. 100685, 2022.
- [32] R. Sanfelice and L. Praly, "Convergence of nonlinear observers on \mathbb{R}^n with a Riemannian metric (Part I)," *IEEE Transactions on Automatic Control*, vol. 57, no. 7, pp. 1709–1722, 2012.
- [33] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, 2023.
- [34] D. Sheng and G. Fazeskas, "A feature learning siamese model for intelligent control of the dynamic range compressor," in *2019 IEEE International Joint Conference on Neural Networks*, 2019, pp. 1–8.
- [35] A. Paszke et al., "Pytorch: an imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [37] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.