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:

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 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 Lorenz 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 \( i = 1, \ldots, N \), we denote with \( \text{col}(A_1, \ldots, A_N) \) the matrix \( (A_1) \mathcal{I}_{n_1 \times n_1} \cdots (A_N) \mathcal{I}_{n_N \times n_N} \) and with \( \text{diag}(A_1, \ldots, A_N) \) the square matrix with \( A_1, \ldots, A_N \) 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

\[
L_f P(x,t) := P(x,t) \frac{\partial f}{\partial x}(x,t) + \frac{\partial f}{\partial t}(x,t) P(x,t) + \partial f P(x,t)
\]

\[
\partial f P(x,t) := \lim_{h \to 0} \frac{P(X(x,t+h,t)\),t) - P(X(x,t))}{h} + \partial P(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 \( X(x,t,t_0) = x \), 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 $P : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$), 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. $L_g P(x,t) = 0$ for all $i = 1, \ldots, m$, 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 $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ in which $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a set of $N \in \mathbb{N}$ vertices (or nodes), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 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 $w_{kj} \geq 0$ of the adjacency matrix $A \in \mathbb{R}^{N \times N}$. We denote by $L \in \mathbb{R}^{N \times N}$ the Laplacian matrix of the graph, defined as $L_{ij} = -w_{ij}$ for $i \neq j$, and $L_{ii} = \sum_{j=1}^{N} w_{ji}$ for $i = j$, where $w_{ij}$ is the $(j,k)$-th entry of $L$. We denote with $N_i$ the set of neighbors of node $i$, i.e. the set $N_i := \{j \in \{1, \ldots, N\} \mid e_{ji} \in \mathcal{E}\}$.

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. $N_1 = \emptyset$. 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 $i,j = 2, \ldots, N$), 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 $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is undirected and leader-connected. Then the Laplacian $L$ can be partitioned as $L = \begin{bmatrix} 0 & \mathcal{L}_{12} \\ \mathcal{L}_{21} & \mathcal{L}_{22} \end{bmatrix}$. Moreover, there exists a strictly positive real number $\mu > 0$ such that $\mathcal{L}_{22} \geq \mu I$.

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 are described by

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

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

$$\dot{x}_i = f(x_i,t) + g(x_i,t)u_i, \quad y_i = h(x_i,t), \quad i = 2, \ldots, N,$$

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 $x := (x_1, \ldots, x_N) \in \mathbb{R}^{Nn}$.

Furthermore, we denote with $\mathcal{X}_i(x^0, t, t_0)$ the trajectory of agent $i$ evaluated at time $t \geq t_0$ such that $\mathcal{X}_i(t_0) = x^0$. The vector field $g$ is chosen to be $
abla(x^0, t, t_0)$ the trajectory of the entire network (2) evaluated at initial condition $x^0 \in \mathbb{R}^{Nn}$, initial time $t_0 \in \mathbb{R}$ at all time $t \geq t_0$. Our synchronization objective is to design a nonlinear diffusive coupling, namely a distributed feedback control law of the form

$$u_i = \sum_{j \in N_i} a_{ij} [\varphi(x_i, y_j, t) - \varphi(x_i, y_i, t)] - \sum_{j=1}^{N} \ell_{ij} \varphi(x_i, y_j, t)$$

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

$$D := \{x \in \mathbb{R}^{Nn} \mid x_i = x_1, \text{ for all } i \in \{1, \ldots, N\}\},$$

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 $j \in N_i$ 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 $D$ defined in (4) is globally uniformly exponentially stable for the closed-loop dynamics

$$\dot{x}_i = f(x_i,t) + g(x_i,t)u_i, \quad i = 2, \ldots, N,$$

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

$$|\mathcal{X}(x^0, t, t_0)|_D \leq k \exp(-\lambda(t-t_0))|x^0|_D, \quad \forall t \geq t_0.$$

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

Assumption 1 The graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is undirected and leader-connected. Moreover, for each $(x^0_1, t_0)$ 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

$$\dot{x}_1 = f(x_1,t), \quad \dot{x}_i = f(x_i,t) + g(x_i,t)u_i, \quad i = 2, \ldots, N.$$  

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 $\rho, \tilde{\rho}, \rho, \lambda > 0$ such that the following holds for all $(x,t) \in \mathbb{R}^{n} \times \mathbb{R}$

$$L_f P(x,t) - \rho P(x,t)g(x,t)g(x,t)^T P(x,t) \leq -2\lambda P(x,t),$$

$$P I \preceq P(x,t) \preceq \tilde{P} I.$$
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 $|x| = \sqrt{x^\top P x}$. 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) $PA + A^\top P - \rho PBB^\top P \preceq -2\lambda P$ which admits a solution under the mild assumption that $(A, B)$ is stabilizable. In this case, a stabilizing control action is given by $u = -\kappa B^\top P x$ 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 $P g$ satisfies an integrability condition in the sense that, denoting $g = [g_1 \ldots g_m]$, there exists a $C^2$ function $\alpha = (\alpha_1, \ldots, \alpha_m)$, $\alpha_i : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ for $\iota = 1, \ldots, m$, satisfying

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

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 $\alpha$ 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 $D$ 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 $G = (V, E, A)$ of agents (1) and let Assumptions 1 to 4 hold. Then, for any $\kappa \geq \rho\|P\|_{\infty}$, with $\rho$ 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 $\alpha$ 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

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

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

$$\dot{\tilde{x}}_i = f(\tilde{x} + \tilde{x}_i, t) - f(z, t)$$

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

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

$$\dot{\tilde{x}}_i = f(\tilde{x} + \tilde{x}_i, t) - f(z, t)$$

$$- \kappa g(\tilde{x} + \tilde{x}_i, t) \sum_{j=2}^{N} \ell_{ij} \left[ \alpha(z + \tilde{x}_j, t) - \alpha(z, t) \right]. \quad (12)$$

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

Now, given $(z^0, \tilde{x}^0, \eta_0) \in \mathbb{R}^{Nn} \times \mathbb{R}$, let $T > t_0$ be the time of existence of the solution of (12) initialized from $(z^0, \tilde{x}^0)$ at time $t_0$. For $t \in [t_0, T)$, let $(Z(t), \tilde{X}(t))$ denotes this solution. Consider the function $\Gamma : [0, 1] \times [t_0, T] \to \mathbb{R}^{Nn}$, with $\Gamma = (\Gamma_2, \ldots, \Gamma_N)$ which satisfies $\Gamma(s, t_0) = s \tilde{x}^0$, and where $\Gamma_i, i = 2, \ldots, N$, is the solution of the following ordinary differential equation for $t_0 \leq t < t_0 + T$

$$\frac{\partial \Gamma_i}{\partial t}(s,t) = g(\zeta_i(s,t), t) - f(Z(t), t)$$

$$- \kappa g(\zeta_i(s,t), t) \sum_{j=2}^{N} \ell_{ij} \left( \alpha(\zeta_j(s,t), t) - \alpha(Z(t), t) \right))$$

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

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

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 $(k, l) \in \{1, \ldots, n\}^2$ we have

$$\frac{d}{dt}[P(\zeta_k(s,t), t, kl)] = \frac{\partial P_{kl}(\zeta_k(s,t), t)}{\partial x} \frac{\partial \zeta_k}{\partial t}(s,t) + \frac{\partial P_{kl}(\zeta_k(s,t), t)}{\partial \zeta_k} \frac{\partial \zeta_k}{\partial t}(s,t),$$

$$= \frac{\partial P_{kl}(\zeta_k(s,t), t)}{\partial x} [f(Z(t) + \frac{\partial \Gamma_k}{\partial t}(s,t))] + \frac{\partial P_{kl}(\zeta_k(s,t), t)}{\partial \zeta_k} [\frac{\partial \Gamma_k}{\partial t}(s,t)].$$
This implies that for all vector $\nu$ in $\mathbb{R}^n$, and $i = 2, \ldots, N$,
\[
\frac{d}{dt} \left[ \nu^T P(\zeta_i(s, t), t) \nu \right] = \nu^T \partial_f P(\zeta_i(s, t), t) \nu \\
- \kappa \sum_{j=2}^N \left[ \ell_{ij} m \nu^T \partial_y P(\zeta_i(s, t), t) \nu \right.
\]
\[
\times \left. (\alpha_i \zeta_j(s, t), t) - \alpha_i (Z(t), t)) \right].
\]

By using the Killing vector assumption (9) and the integrability one in (10), the time derivative of $V_i$ becomes
\[
\dot{V}_i(t) = \int_0^1 \left[ \partial_{\Gamma_i}^T(s, t) L_f P(\zeta_i(s, t), t) \partial_{\Gamma_i}(s, t) \\
- 2\kappa \partial_{\Gamma_i}^T(s, t) \sum_{j=2}^N \ell_{ij} P(\zeta_i(s, t), t) g(\zeta_i(s, t), t) \\
\times g(\zeta_i(s, t), t)^T P(\zeta_i(s, t), t) \frac{\partial_\nu^T}{\partial s}(s, t) \right] ds.
\]

With the following notations,
\[
D(s, t) := \text{diag} \{ L_f P(\zeta_i(s, t), t) \}_{i=2, \ldots, N} \\
\Psi(s, t) := \text{col} \{ P(\zeta_i(s, t), t) g(\zeta_i(s, t), t) \partial_{\Gamma_i}(s, t) \}_{i=2, \ldots, N},
\]
we compute the derivative of $V$ as follows
\[
\dot{V}(t) = \int_0^1 \left[ \partial_{\Gamma_i}^T(s, t) D(t) \partial_{\Gamma_i}(s, t) - 2\kappa \Psi(s, t) L_{222} \Psi(s, t)^T \right] ds
\]
\[
\leq \int_0^1 \left[ \partial_{\Gamma_i}^T(s, t) D(t) \partial_{\Gamma_i}(s, t) - 2\kappa \rho \Psi(s, t) \Psi(s, t)^T \right] ds
\]
where in the second step we used Assumption 1 and Lemma 1. Therefore, by selecting $\kappa \geq \frac{\rho}{2\mu}$ with $\rho$ satisfying inequality (8) and $\mu > 0$ given by Lemma 1, we get $V(t) \leq -\lambda V(t)$. From Gronwall’s Lemma, this implies $V(t) \leq \exp(-\lambda(t-t_0)) V(t_0)$ for all $t \in [t_0, t_0 + T]$. Moreover, with (13) it yields
\[
V_i(t) \geq \frac{p}{2} \int_0^1 \partial_{\Gamma_i}^T(s, t) \partial_{\Gamma_i}(s, t) ds \geq \frac{p}{2} ||\dot{\chi}_i(t)||^2.
\]
Moreover, since
\[
V(t_0) \leq \frac{p}{2} ||\dot{\chi}(t_0)||^2,
\]
it yields for all $t \in [t_0, t_0 + T]$ that
\[
||\dot{\chi}(t)||^2 \leq \exp(-\lambda(t-t_0)) \frac{p}{2} ||\dot{\chi}(t_0)||^2.
\]
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. $T = +\infty$). Moreover, by equivalence of norms in finite dimensional spaces it follows that there exist two strictly positive real numbers $\xi, \tau > 0$ such that $\xi ||x(t, t_0)|| \leq ||x(t)|| \leq \xi ||x(t, t_0)||$, 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 case to the network of agents described as
\[
\begin{align*}
\dot{x}_1 &= f(x_1, t), & y_1 &= h(x_1, t), \\
\dot{x}_i &= f(x_i, t) + u_i, & y_i &= h(x_i, t),
\end{align*}
\]
for all $i = 2, \ldots, N$, 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 $\bar{p}, \bar{p}, \lambda, \rho > 0$ such that the following holds for all $(x, t) \in \mathbb{R}^n \times \mathbb{R}$
\[
\begin{align*}
L_f P(x, t) - \rho \frac{\partial h}{\partial x}(x, t)^T \frac{\partial h}{\partial x}(x, t) &\leq -2\lambda P(x, t), \\
\bar{p} I &\leq P(x, t) \leq \bar{p} I.
\end{align*}
\]

**Assumption 6** The vector field $\beta : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ defined as
\[
\beta(x, t) = P(x, t)^{-1} \frac{\partial h}{\partial x}(x, t)^T
\]
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 $PA + A^T - \rho C^T C \preceq -2\lambda P$, provided a control gain of the form $\varphi(x_i, y_i, t) = -P^{-1} C^T y_i$. 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 $G = \{V, E, \mathcal{A}\}$ of agents (18) and suppose Assumptions 1, 5 and 6 hold. Then, for any $\kappa \geq \frac{\rho}{2\mu}$, with $\mu$ given by Lemma 1, the distributed output-feedback control law (3) with $\varphi(x_i, y_i, t) = \kappa \beta(x_i, t)$ 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
\[
\begin{align*}
\dot{x}_1 &= f(x_1, t), && y_1 = h(x_1, t), \\
\dot{x}_i &= f(x_i, t) + g(x_i, t) u_i, && y_i = h(x_i, t),
\end{align*}
\]
for all $i = 2, \ldots, N$ and $y_i, u_i \in \mathbb{R}^m$. 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 $(x, t) \in \mathbb{R}^n \times \mathbb{R}$
\[
P(x, t) g(x, t) = \frac{\partial h}{\partial x}(x, t)^T, \quad L_g P(x, t) = 0.
\]

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 $PA + A^T - \rho C^T C \preceq -2\lambda P$, with $PB = C^T$. Finally, we have then the following result. The proof is omitted for space reasons.

**Proposition 2** Consider a network $G = \{V, E, \mathcal{A}\}$ of agents (20) and suppose Assumptions 1, 5 and 7 hold. Then, for any $\kappa \geq \frac{\rho}{2\mu}$, with $\mu$ given by Lemma 1, the distributed output-feedback control law (3) with $\varphi(x_i, y_i, t) = \kappa y_j$ 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 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 $\alpha : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ and a scalar $\varepsilon > 0$ such that, for $i = 1, \ldots, m$, the following holds

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

**Theorem 2** Consider a network $G = \{V, E, A\}$ of agents (1) and let Assumptions 1, 2, 3 and 4’ hold. Moreover, assume there exists a positive real number $\bar{\gamma}$ such that $|g_i(x, t)| \leq \bar{\gamma}$ for all $(x, t)$ in $\mathbb{R}^n \times \mathbb{R}$ and $i = 1, \ldots, m$. Let $\mu$ be given by Lemma 1 and let $L = \max_{i,j} |\ell_{ij}|$ where $|\ell_{ij}|$ is the Laplacian matrix associated to the graph. Then, if $\varepsilon$ in Assumption 4 is selected as $\varepsilon \in [0, \varepsilon^*]$ with $\varepsilon^* = \frac{\lambda \mu p}{\rho N \mu \bar{\gamma}^2}$, there exists $\kappa^*$ such that for any $\kappa \in [\frac{\mu p}{\rho N \mu \bar{\gamma}^2}, \kappa^*]$, the distributed state-feedback control law (3)-(15) solves the synchronization Problem 1.

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

$$\dot{V}_i(t) = \int_{t_0}^{t} \left[ \frac{\partial \Gamma_i^\top}{\partial s} D + \frac{\partial \Gamma_i^\top}{\partial s} - 2\kappa \mu P \Gamma + T_i \right] ds,$$

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

$$T_i(s, t) = -2\kappa \frac{\partial \Gamma_i}{\partial s}(s, t)^\top \sum_{j=1}^{N} \sum_{i=1}^{m} \ell_{ij} \sum_{i=1}^{m} P(\zeta_j(s, t), t) g_i(\zeta_i(s, t), t) \times \left( \frac{\partial g_i}{\partial x}(\zeta_j(s, t), t) - g_i(\zeta_i(s, t), t) P(\zeta_i(s, t), t) \right) \frac{\partial \Gamma_j}{\partial s}(s, t).$$

With Assumption 4’, the bound on $P$, $g$ and the $\ell_{ij}$’s, the term $T_i$ can be bound as follows

$$T_i(t, s) \leq 2\kappa \left\| \frac{\partial \Gamma_i}{\partial s}(s, t, t_0) \right\| N \bar{\gamma} \| P \| \| \frac{\partial \Gamma_i}{\partial s}(s, t) \|$$

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

$$\dot{V}(t) \leq \int_{t_0}^{t} \left[ \frac{\partial \Gamma}{\partial s}(s, t)^\top D(s, t) \frac{\partial \Gamma^\top}{\partial s}(s, t) - 2\kappa \mu P(s, t) \Gamma(s, t) + 2\kappa NL \bar{\gamma} \| P \| \| \frac{\partial \Gamma}{\partial s}(s, t) \| \right] ds.$$

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,

$$\dot{V}(t) \leq \int_{t_0}^{t} \left[ \frac{\partial \Gamma}{\partial s}(s, t)^\top \text{diag}(\mathcal{T}) \frac{\partial \Gamma}{\partial s}(s, t) \right] ds,$$

where $\mathcal{T} = L \frac{P}{2} + 2\kappa \mu P - 2\kappa \mu P \gamma \gamma^\top P$, with the functions $P$ and $g$ depending on $(\zeta, t)$ and $\bar{\gamma} = \frac{N L \bar{\gamma}}{\rho N \mu \bar{\gamma}^2}$. With Assumption 2, this implies $\mathcal{T} \leq (2\kappa \bar{\gamma} - \lambda P + (\rho - 2\kappa \mu) \gamma \gamma^\top P)$. Note that with the choice of $\varepsilon$ in the statement of the theorem, we have $\frac{\rho}{\lambda} < \frac{\kappa}{\lambda}$. Consequently, for each $\kappa \in \left[ \frac{\mu p}{\rho N \mu \bar{\gamma}^2}, \frac{\mu p}{\rho N \mu \bar{\gamma}^2} \right]$ it implies $\dot{V}(t) \leq -\lambda \dot{V}(t)$ for all $t \geq t_0$, where $\lambda = 2\kappa \bar{\gamma} - \lambda$ is a positive real number. The proof follows completing 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 $G = \{V, E, A\}$ of systems (1). Suppose Assumption 1, 2, and 4 hold. Let $\kappa \geq \frac{\mu p}{\rho N \mu \bar{\gamma}^2}$ be fixed. Assume $|g(x)| \leq \bar{\gamma}$ for all $x$ in $\mathbb{R}^n$. Then, for each $x > 0$ there exist $k, \varepsilon > 0$ such that, if the following holds

$$|L_g P(x, t)| \leq k, \quad (t, x, t) \in \mathbb{R}^n \times \{t_0, \infty\} \times \{1, \ldots, m\}, \tag{23}$$

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

$$|\mathcal{X}(x_\lambda^*, t, t_0)|_{\mathcal{D}} \leq k \exp(-\frac{\lambda}{2} (t-t_0)) |\mathcal{X}(x_\lambda^*, t_0)|_{\mathcal{D}}, \forall t \geq t_0. \tag{24}$$

**Proof:** Let $x, t_0 > 0$ and let $x^* \in \mathbb{R}^{Nn}$ satisfy $\|x^*\|_{\mathcal{D}} \leq k$. Assume that (23) is satisfied for some positive real number $\varepsilon$ 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 $(\varepsilon_\mathcal{V}, \varepsilon_V)$ such that

$$\varepsilon_\mathcal{V} |\mathcal{X}(x_\lambda^*, t)|_{\mathcal{D}}^2 \leq V(t) \leq \varepsilon_V |\mathcal{X}(x_\lambda^*, t)|_{\mathcal{D}}^2.$$
Employing that \( f_0^1 \| \frac{\partial V_j}{\partial s} (s,t) \| ds \leq \frac{V_j(t)}{P} \), and with (23), this gives

\[
\dot{V}_i(t) = \int_0^1 \left[ \frac{\partial \dot{V}}{\partial s} D \frac{\partial V}{\partial s} - 2\kappa \rho \Psi \dot{V} + V_i(t) \varepsilon_{i\hat{e}e} \left( 1 + \frac{V_j(t)}{P} \right) \right] ds
\]

where \( \varepsilon = mN \ell \bar{L} \hat{g} \) where \( \bar{L} = \max |\ell_{ij}| \). Consequently, following the proof of Theorem 1, we obtain, with \( \varepsilon \geq \frac{\rho P}{m} \),

\[
\dot{V} \leq - (\lambda - \varepsilon_{i\hat{e}e}) V + \varepsilon_{i\hat{e}e} \sum_{i=2}^N V_i^2 \leq - (\lambda - \varepsilon_{i\hat{e}e} - \varepsilon_{i\hat{e}e} V). 
\]

Finally, selecting \( \varepsilon < \frac{\lambda}{\varepsilon_{i\hat{e}e}} \) it implies \( \dot{V} \leq - \frac{\lambda}{\varepsilon_{i\hat{e}e}} V \) 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 expressiveness 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

\[
P(x, \theta^\prime) = \begin{bmatrix} P_1(x, \theta^\prime) & \cdots & P_n(x, \theta^\prime) \\ \vdots & \ddots & \vdots \\ P_n(x, \theta^\prime) & \cdots & P_n(x, \theta^\prime) \end{bmatrix},
\]

where \( p = \binom{n(n+1)}{2} \) is the total number of entries to be learned, the vector \( g = (P_1(x, \theta^\prime), \ldots, P_n(x, \theta^\prime))^{\top} \) is the output of the neural network DNN\(_P : \mathbb{R}^n \times \mathbb{R}^{n^2} \rightarrow \mathbb{R}^p \) and \( \theta^\prime \in \mathbb{R}^{n^2} \) 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 \( g(x)^\top P(x, \theta^\prime) \) 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

\[
J_P(x, \theta^\prime) = \sum_{i=1}^4 w_i J_i(x, \theta^\prime),
\]

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

\[
J_i(x, \theta^\prime) = \ln \left( \max \{9 \bar{\Xi}(\lambda_M (M_i)), 0 \} + 1 \right),
\]

with \( i = 1, \ldots, 4 \), \( \lambda_M \) being the maximum eigenvalue and \( M_i \) defined as

\[
M_1 = L_f P(x, \theta^\prime) - \rho P(x, \theta^\prime) g(x)^\top (x) P(x, \theta^\prime) + \epsilon I, \\
M_2 = L_g P(x, \theta^\prime) - \epsilon I, \\
M_3 = -L_g P(x, \theta^\prime) - \epsilon I, \\
M_4 = -P(x, \theta^\prime) + \beta I
\]

where \( \rho, \epsilon, \beta > 0 \) are positive scalars with \( \epsilon > 0 \). And where \( \bar{\Xi}(\lambda) \) \( \in \mathbb{C}^n \) 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 \( \rho, \epsilon, \beta \). 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 \( \theta'' \in \mathbb{R}^{n \times n'} \) of the second network \( \text{DNN}_\alpha : \mathbb{R}^n \times \mathbb{R}^{n'} \rightarrow \mathbb{R}^m \) such that
\[
J_\alpha(x, \theta'') = \left( \frac{\partial \text{DNN}_\alpha(x, \theta'')}{\partial x} - g(x) \right)^T P(x, \theta') \left( \frac{\partial \text{DNN}_\alpha(x, \theta'')}{\partial x} - g(x) \right)
\]
is minimized. The full learning procedure is summarized by Algorithm 1. Clearly, the DNNs can be trained only on a dataset \( \mathcal{D} \) 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 \( \mathcal{D} \subset \mathbb{R}^n \) a set with a finite number of elements and let \( r > 0 \) be such that
\[
S \subseteq \bigcup_{x_i \in \mathcal{D}} B(x_i, r), \quad B(x_i, r) := \{x \in \mathbb{R}^n : |x - x_i| < r\}.
\]

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 \( M(x) \preceq -2qI \) for all \( x_i \in \mathcal{D} \) 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
\[
M(x) = M(x_i) + M(x) - M(x_i) \preceq M(x_i) + |M(x) - M(x_i)|I
\leq M(x_i) + L_M|x - x_i|I \preceq -q \left( \frac{2 - |x - x_i|}{r} \right) I
\]
for an arbitrary \( x_i \in \mathcal{D} \). Then, \( M(x) \preceq -qI \) for all \( x \in B(x_i, r) \).

The result follows from the fact that \( S \subseteq \bigcup_{x_i \in \mathcal{D}} B(x_i, r) \).

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 \( \alpha \).

Since the estimated metric is a DNN and \( g \in C^2 \), their product is Lipschitz continuous on a compact set. Since \( \alpha \) 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 problem\(^1\). We consider a network of \( N = 6 \) identical Lorenz attractors. Such systems are particularly interesting, since they can present a chaotic behavior. Each agent \( i = 1, \ldots, N \) is described by the following dynamics
\[
\begin{align*}
\dot{x}_{i,1} &= a(x_{i,2} - x_{i,1}) + u_i \\
\dot{x}_{i,2} &= x_{i,1}(b - x_{i,3}) - x_{i,2} + (2 + \sin(x_{i,1}))u_i \\
\dot{x}_{i,3} &= x_{i,1}x_{i,2} - cx_{i,3}
\end{align*}
\]
with \( x_i = (x_{i,1}, x_{i,2}, x_{i,3}) \in \mathbb{R}^3 \) and where \( a = 10, b = 28, c = 8/3 \), 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 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 \( 3 \times 10^{-3} \), while DNN\( \alpha \) uses a learning rate of \( 5 \times 10^{-3} \). 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 \( 2 \times 10^5 \) 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., \( u_i = \varphi(x_{i,1} + \nu_i) \) 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 \( g(x)^T P(x, \theta') \). 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.

---

\(^1\)The code for reproducing the experiments proposed in this section can be found at https://github.com/SmamieZoboli/Control-learning-multiagent-lorenz.git
Lectures in feedback design for multivariable systems.

REFERENCES


