

# Unconstrained Parametrization of Dissipative and Contracting Neural Ordinary Differential Equations

Daniele Martinelli, Clara Lucía Galimberti, Ian R. Manchester, Luca Furieri, and Giancarlo Ferrari-Trecate

**Abstract**—In this work, we introduce and study a class of Deep Neural Networks (DNNs) in continuous-time. The proposed architecture stems from the combination of Neural Ordinary Differential Equations (Neural ODEs) with the model structure of recently introduced Recurrent Equilibrium Networks (RENs). We show how to endow our proposed NodeRENs with contractivity and dissipativity — crucial properties for robust learning and control. Most importantly, as for RENs, we derive parametrizations of contractive and dissipative NodeRENs which are unconstrained, hence enabling their learning for a large number of parameters. We validate the properties of NodeRENs, including the possibility of handling irregularly sampled data, in a case study in nonlinear system identification.

## I. INTRODUCTION

Learning complex nonlinear mappings from data is a fundamental challenge in various engineering applications, including computer vision, healthcare, internet of things, and smart cities [1]. Deep Neural Networks (DNNs) have emerged as powerful tools for this task, thanks to their adaptability and ability to generalize their predictions from large amounts of data. However, DNNs often lack robustness: a slight change in the input data may yield highly different outputs, eventually leading to poor generalization capabilities [2]–[4]. This deficiency becomes especially critical when DNNs are implemented in real-world systems like safety-critical power grids or human-interacting robots. In these applications, sensor measurements are inevitably affected by multiple sources of noise and uncertainty, and a lack of robustness may result in substantial losses.

To endow DNN models with formal stability and robustness guarantees, [3] proposes to equip DNN layers with a dynamical system interpretation. The work [5] establishes dynamical DNN models which universally approximate all nonlinear dynamical systems defined in discrete-time and proposes a set of convex constraints that enforce the stability of the DNN model during training. In [6], the authors have developed discrete-time Recurrent Equilibrium Networks (RENs) that result from the closed-loop interconnection of a discrete-time linear dynamical system with a static nonlinearity. A main contribution of [6] is to

provide an unconstrained parametrization (also known as *direct* parametrization) of a class of RENs with properties of stability and dissipativity that are *built-in*, i.e., that holds for any choice of the parameters, and without the need to constrain them to a subset. This property enables parameter optimization for very deep models through unconstrained-gradient-descent-based algorithms, while ensuring stability and dissipativity at any iteration. The usefulness of RENs for system identification and optimal control has been demonstrated in [6] and [7], respectively. However, the corresponding stability and dissipative guarantees are only compatible with discrete-time or sampled-data systems. Studying how the properties of nonlinear discrete-time dynamics port to their continuous-time counterpart is usually a challenging problem [8].

The recently proposed Neural Ordinary Differential Equations (Neural ODEs) [9] offer a bridge between DNNs and continuous-time dynamics. By unfolding infinitely many DNN layers using a parametrized ODE and employing the *adjoint sensitivity method* for training [9], Neural ODEs offer several advantages over discrete DNN models. These include memory efficiency, adaptive computations beyond Euler-like integration methods, and the ability to handle data arriving at arbitrary time instants.

Expanding on Neural ODEs, [10] introduced a continuous-time analogous of Recurrent Neural Networks, albeit without stability or dissipativity guarantees. The work [11] has proposed stable Hamiltonian Neural ODEs which further exhibit non-vanishing gradient flows. Additionally, [4] developed a class of Hamiltonian ODEs with contractivity properties. Furthermore, [12] derived architectures ensuring convergence to a stable set. However, an architecture combining the robustness and expressiveness of [5], [6] with the benefits of Neural ODEs is not available to the best of our knowledge.

### A. Contributions

In this paper, we establish an alternative version of the REN architecture of [6] that is compatible with Neural ODEs in continuous-time. Consequently, we call our architectures *NodeRENs*. After showing that NodeRENs induce well-posed dynamical systems in continuous-time, we establish that our models enjoy stability and dissipativity properties by design, i.e., without any constraints on the space of the parameters that describe the DNNs — akin to their discrete-time counterpart [6]. NodeRENs, also inherit the advantages of Neural ODEs in continuous-time; they are compatible with any integration scheme that can be chosen based on a trade-off between accuracy and computational resources, and

D. Martinelli, C. L. Galimberti, L. Furieri, and G. Ferrari-Trecate are with the Institute of Mechanical Engineering, EPFL, Switzerland. E-mail addresses: {daniele.martinelli, clara.galimberti, luca.furieri, giancarlo.ferraritrecate}@epfl.ch.

Ian Manchester is with the Australian Centre for Robotics (ACFR) and the School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, Australia. E-mail address: ian.manchester@sydney.edu.au.

Research supported by the Swiss National Science Foundation (SNSF) under the NCCR Automation (grant agreement 51NF40.80545). Luca Furieri is also grateful to the SNSF for the Ambizione grant PZ00P2.208951.

they can be evaluated at any chosen time instant, without the requirement to be uniformly sampled in time.

The paper is structured as follows. In Section II, we describe the problem setting, including the definition of contractivity and Integral Quadratic Constraints (IQCs). In Section III, we introduce the NodeREN model and detail the steps to obtain NodeRENs that are contracting and dissipative by design. Moreover, in Section IV the properties and performance of NodeRENs are validated on a nonlinear system identification problem. Finally, in Section V we summarize the conclusions of this work, outlining potential future directions for research.

For the sake of conciseness, the proofs of all theorems in this work are reported in the Appendix of the full version [13].

### B. Notation

We denote the set of non-negative real numbers as  $\mathbb{R}_0^+$ . For  $T > 0$ , let  $PC([0, T], \mathbb{R}^n)$  be the space of piecewise-continuous functions in the time interval  $[0, T]$ . We represent the Euclidean norm of  $v \in \mathbb{R}^n$  with  $|v|$ . For a square matrix  $X$ , we use the notation  $X \succ 0$  ( $X \succeq 0$ ) and  $X \prec 0$  ( $X \preceq 0$ ) to denote positive (semi-) definiteness and negative (semi-) definiteness, respectively. The minimum and maximum eigenvalues of the square matrix  $X$  are denoted as  $\lambda_{\min}(X)$  and  $\lambda_{\max}(X)$ , respectively. We use  $(*)$  to represent a symmetric term in a quadratic expression, e.g.,  $(X+Y)Q(*)^\top = (X+Y)Q(X+Y)^\top$ , for some  $Q \in \mathbb{R}^{n \times n}$  and  $X, Y \in \mathbb{R}^{q \times n}$ . With  $(*)$ , we also indicate elements in symmetric matrices that can be obtained by symmetry. We use  $[M]_{p \times m}$  to indicate the block matrix with the first  $p$  rows and  $m$  columns of  $M \in \mathbb{R}^{n \times n}$  with  $n \geq p$  and  $n \geq m$ .

## II. PROBLEM FORMULATION

Consider a nonlinear system  $\Sigma_\theta$  in continuous-time

$$\Sigma_\theta = \begin{cases} \dot{x}(t) = f_\theta(x(t), u(t)) \\ y(t) = g_\theta(x(t), u(t)) \end{cases}, \quad (1)$$

where  $x(t) \in \mathbb{R}^n$ ,  $y(t) \in \mathbb{R}^p$  and  $u(t) \in \mathbb{R}^m$  denote the state, output and input of the system at any time  $t \in \mathbb{R}_0^+$ , respectively, and  $x(0) = x_0$ . In (1),  $\theta \in \mathbb{R}^{n_\theta}$  denotes a vector of parameters affecting the behavior of the system. The idea behind Neural ODEs [9] is to interpret some specific DNNs (e.g., ResNets [14]) as discretized ODEs. Then, the family of Neural ODEs can be obtained when the discretization step converges to zero, effectively passing from discrete-time DNNs to continuous-time models as (1).<sup>1</sup> Accordingly, in this paper, we consider the following learning problem:

$$\begin{aligned} \min_{\theta} \quad & L(\theta, \mathcal{Z}) \\ \text{subject to} \quad & (1). \end{aligned} \quad (2)$$

$L$  is a scalar loss function that can depend on the trajectories  $x(t)$  and  $y(t)$  of (1), for  $t \in [0, T]$  with  $T > 0$ , and  $\mathcal{Z}$  is a

<sup>1</sup>For instance, standard ResNet models whose hidden states evolve as  $h_{k+1} = h_k + \Delta t f(h_k, \theta_k)$  can be interpreted as the ODE  $\dot{h}(t) = f(h(t), \theta(t))$  as the discretization step  $\Delta t$  tends to zero.

given training dataset. For instance, in a classification task,  $y(T)$  can play the role of the DNN output layer which is compared to the label  $y_{label}^j$  of the point  $x^j(0)$  contained in a dataset  $\mathcal{Z} = \left\{ (x^j(0), y_{label}^j) \right\}_{j=1}^N$  for  $N \in \mathbb{N}$ . Neural ODEs can exploit any chosen numerical method to perform the forward propagation and the gradient computations through (1), including those based on adaptive sampling to guarantee a desired level of precision (e.g., Dormand-Prince, Bogacki–Shampine [15]). Although possible, back-propagating through numerical solver’s operations can be highly expensive in terms of memory and introduce numerical errors. As an alternative, to solve (2) one can use the adjoint method — we refer to [9] for full details.

Neural-ODEs in their general form are not guaranteed to yield stable or dissipative dynamical flows. Such properties are fundamental in optimal control and system identification, as well as in robust learning problems dealing with noisy features [4] and adversarial attacks [16]. Specifically, motivated by [6], in this paper, we focus on continuous-time systems that are contracting and systems that satisfy incremental IQCs. We proceed with formally defining both.

Firstly, given a system  $\Sigma_\theta$  with initial condition  $a \in \mathbb{R}^n$  and an input function  $u^{[1]} \in PC([0, \infty], \mathbb{R}^m)$ , let  $x_a^{[1]}$  and  $y_a^{[1]}$  denote the corresponding state and output trajectories, respectively.

**Definition 1.** A system  $\Sigma_\theta$  in the form (1) is said to be *contracting* if for any two initial conditions  $a, b \in \mathbb{R}^n$  and given the same input trajectory  $u^{[1]} \in PC([0, \infty], \mathbb{R}^m)$ , the corresponding state trajectories  $x_a^{[1]}$  and  $x_b^{[1]}$  satisfy:

$$|x_a^{[1]}(t) - x_b^{[1]}(t)| \leq \kappa e^{-ct} |a - b|, \quad (3)$$

for all  $t \in \mathbb{R}_0^+$  and for some  $c > 0$ ,  $\kappa > 0$ .

The Definition 1 can be interpreted as follows: a contracting system ‘forgets’ the initial condition exponentially fast as time progresses. Hence, all trajectories converge to each other, independently of the initial state.

Next, we define dissipative systems that satisfy incremental IQCs. Dissipative systems cannot increase their internal energy despite external inputs (e.g., feedback control actions or disturbances). As such, they can be designed to possess finite input-output gains, a crucial property in nonlinear control theory [17], as well as robust learning [18]. In order to formally define these properties, let  $a, b \in \mathbb{R}^n$  be two initial conditions and  $u^{[1]}, u^{[2]} \in PC([0, \infty], \mathbb{R}^m)$  be two input trajectories. Define the relative displacements as

$$\begin{aligned} \Delta y(t) &= y_a^{[1]}(t) - y_b^{[2]}(t), \quad \Delta u(t) = u^{[1]}(t) - u^{[2]}(t), \\ \Delta x(t) &= x_a^{[1]}(t) - x_b^{[2]}(t). \end{aligned} \quad (4)$$

Let

$$s_\Delta(\Delta u(t), \Delta y(t)) = \begin{bmatrix} \Delta y(t) \\ \Delta u(t) \end{bmatrix}^\top \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix} \begin{bmatrix} \Delta y(t) \\ \Delta u(t) \end{bmatrix}, \quad (5)$$

be a quadratic function  $s_\Delta : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$  parametrized by  $Q \in \mathbb{R}^{p \times p}$ ,  $S \in \mathbb{R}^{m \times p}$ ,  $R \in \mathbb{R}^{m \times m}$ . We recall the

definition of dissipative systems that satisfy IQCs according to the supply rate (5).

**Definition 2.** A system  $\Sigma_\theta$  in form (1) satisfies the *incremental* IQC defined by the matrices  $(Q, S, R)$ , with  $Q \preceq 0$  and  $R = R^\top$ , if there exists a function  $\mathcal{S} : \mathbb{R}^n \rightarrow \mathbb{R}_0^+$  such that, for any two initial conditions  $a, b \in \mathbb{R}^n$  and any two possible input functions  $u^{[1]}, u^{[2]} \in PC([0, \infty], \mathbb{R}^m)$ ,

$$\mathcal{S}(\Delta x(t_1)) \leq \mathcal{S}(\Delta x(t_0)) + \int_{t_0}^{t_1} s_\Delta(\Delta u(t), \Delta y(t)) dt, \quad (6)$$

for every  $t_1 \geq t_0$ , where  $\Delta x$ ,  $\Delta y$  and  $\Delta u$  are defined in (4) and the supply rate  $s_\Delta$  is defined in (5).

Despite being restricted to quadratic supply functions according to (5), IQCs can certify many incremental properties by appropriately selecting the values of  $(Q, S, R)$ . It is worth noting that the assumptions in Definition 2

$$Q \preceq 0, \quad R = R^\top, \quad (7)$$

are fulfilled for several incremental properties of interest, see Table I.

Incremental Property	$Q$	$R$	$S$	$s_\Delta(\Delta u, \Delta y)$
$L_2$ - gain bound ( $\gamma \geq 0$ )	$-\frac{1}{\gamma}I$	$\gamma I$	0	$\gamma \Delta u ^2 - \frac{1}{\gamma} \Delta y ^2$
Passivity	0	0	$\frac{1}{2}I$	$\Delta u^\top \Delta y$
Input Passivity ( $\nu \geq 0$ )	0	$-2\nu I$	$I$	$\Delta u^\top \Delta y - \nu \Delta u ^2$
Output Passivity ( $\varepsilon \geq 0$ )	$-2\varepsilon I$	0	$I$	$\Delta u^\top \Delta y - \varepsilon \Delta y ^2$

TABLE I: Choices of  $(Q, S, R)$  to verify different incremental properties.

To summarize, our goal is to learn the parameters  $\theta$  of a dynamical system (1) — i.e., a Neural ODE — that optimize a given cost as per (2), with the hard constraint that either (3) or (6) (or both) hold. This new requirement (i.e., the hard constraint) must be satisfied for all the parameters  $\theta$  we optimize over. In other words, we consider a form of fail-safe learning, in the sense that the property of the model to be contracting or dissipative must be guaranteed during and after parameter optimization.

**Remark 1.** In solving (2), one must select a numerical integration scheme for simulating the trajectories of (1) to be used in forward propagation and gradient computations. However, the properties of the continuous-time model (1) may not be preserved in the discretized one, in general. While our focus in this paper is on the contractivity and dissipativity properties of the continuous-time model (1), it is worth noting that integration methods preserving contractivity are discussed in [19]. Adapting these methods to IQC properties, or developing new ones, represents an interesting future research direction.

### III. CONTRACTIVITY AND DISSIPATIVITY OF NODERENS

In this section, we present and analyze novel DNN structures that arise from appropriately combining Neural ODEs [9] with RENs [6]. The starting idea is to define the

functions  $f_\theta$  and  $g_\theta$  in (1) through the model utilized for the layer equation in [6]. Specifically, we consider

$$\begin{bmatrix} \dot{x}(t) \\ v(t) \\ y(t) \end{bmatrix} = \overbrace{\begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}}^{\tilde{A}} \begin{bmatrix} x(t) \\ w(t) \\ u(t) \end{bmatrix} + \overbrace{\begin{bmatrix} b_x \\ b_v \\ b_y \end{bmatrix}}^{\tilde{b}}, \quad (8)$$

$$w(t) = \sigma(v(t)), \quad (9)$$

where  $x(t) \in \mathbb{R}^n$ ,  $u(t) \in \mathbb{R}^m$  and  $y(t) \in \mathbb{R}^p$  are respectively the state, the input, and output at time  $t$ . The function  $\sigma(\cdot)$  represents a nonlinear map and it is applied entry-wise. The input and output of  $\sigma(\cdot)$  are  $v(t), w(t) \in \mathbb{R}^q$ , respectively. We denote system (8)-(9) as a NodeREN, and it can be interpreted as an affine time-invariant system in closed-loop with a static nonlinearity  $\sigma(\cdot)$ . In NodeRENS, the set of trainable parameters  $\theta \in \mathbb{R}^{n_\theta}$  consists of the set  $\tilde{A}$  of matrices  $(A, B_1, B_2, C_1, C_2, D_{11}, D_{12}, D_{21}$  and  $D_{22})$  and the set  $\tilde{b}$  of vectors  $(b_x, b_v, b_y)$  in (8). It is worth noting that the model (8)-(9) is highly flexible, as it encompasses many important neural network architectures. For more details, see [6].

The work [6] has established conditions that ensure contractivity and IQC properties of the *discrete-time* dynamics induced by  $\theta$ . However, the same conditions on  $\theta$  fail to ensure these properties for the continuous-time solutions of (8)-(9), in general.

Towards establishing contractivity and IQC-based dissipativity in continuous-time for the model (8)-(9), consider two different possible trajectories of the system, starting from two initial conditions  $a, b \in \mathbb{R}^n$  and two input functions  $u^{[1]}, u^{[2]} \in PC([0, \infty], \mathbb{R}^m)$ . Then, define the *incremental form* of the system (8)-(9)

$$\begin{bmatrix} \Delta \dot{x}(t) \\ \Delta v(t) \\ \Delta y(t) \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \Delta x(t) \\ \Delta w(t) \\ \Delta u(t) \end{bmatrix}, \quad (10)$$

$$\Delta w(t) = \sigma(v_b^{[2]}(t) + \Delta v(t)) - \sigma(v_b^{[2]}(t)), \quad (11)$$

where  $\Delta x$ ,  $\Delta y$  and  $\Delta u$  are defined in (4). Moreover,  $\Delta v(t) = v_a^{[1]}(t) - v_b^{[2]}(t)$ , where  $v_a^{[1]}$  and  $v_b^{[2]}$  are the inputs of  $\sigma(\cdot)$  for each trajectory. Next, we introduce two technical assumptions.

**Assumption 1.** The function  $\sigma(\cdot)$  belongs to  $PC([0, \infty], \mathbb{R})$  and its slope is restricted to the interval  $[0, 1]$ , that is

$$0 \leq \frac{\sigma(y) - \sigma(x)}{y - x} \leq 1, \quad \forall x, y \in \mathbb{R}, \quad x \neq y.$$

It is important to notice that, under Assumption 1,  $\Delta v(t)$  and  $\Delta w(t)$  verify the following inequality

$$\Gamma(t) = \begin{bmatrix} \Delta v(t) \\ \Delta w(t) \end{bmatrix}^\top \begin{bmatrix} 0 & \Lambda \\ \Lambda & -2\Lambda \end{bmatrix} [*] \geq 0, \quad \forall t \in \mathbb{R}, \quad (12)$$

for any diagonal matrix  $\Lambda \succ 0$ . Note that most of the popular activation functions used in the literature, such as the logistic function,  $ReLU(\cdot)$  and  $\tanh(\cdot)$ , satisfy Assumption 1.

**Assumption 2.**  $D_{11}$  in (8) is strictly lower-triangular.

Assumption 2 enforces that each scalar entry of  $\Delta v(t)$  only depends on the ones above it through (10). Hence, it becomes possible to explicitly compute  $\Delta v(t)$ . This simplifies the numerical calculation of the solutions to (10)-(11), while guaranteeing that the model is very expressive thanks to the recursive application of the nonlinearity  $\sigma$  on successive entries of  $\Delta v(t)$ . The case where  $D_{11}$  is not lower-triangular is left for future work. We refer the interested reader to [6] for a discussion on implicit RENs in discrete-time.

In general, an ODE model may not admit a unique solution for a given initial condition and input trajectory [20]. We now proceed to show that (8)-(9) admits a unique solution for any choice of the parameters  $\theta$ .

**Lemma 1.** Let Assumptions 1 and 2 hold. Then, the model (8)-(9) admits a unique solution in time for any  $u(t) \in PC([0, \infty], \mathbb{R}^m)$ ,  $x(0) \in \mathbb{R}^n$  and for any choice of the parameters  $\theta \in \mathbb{R}^{n_\theta}$ .

The proof, reported in [13] for completeness, is based on deriving a global Lipschitz constant by iterating through the nonlinearities defining the vector  $v(t)$ .

#### A. Characterizing Contracting and Dissipative NodeRENs

Here, we derive sufficient conditions over the parameters  $\theta$  to guarantee contractivity and dissipativity of NodeRENs (8)-(9) according to a specified supply rate (5).

**Theorem 1.** A NodeREN (8)-(9) is contracting according to (3), if there exists a matrix  $P \succ 0$  and a diagonal matrix  $\Lambda \succ 0$  such that

$$\begin{bmatrix} -A^\top P - PA & -C_1^\top \Lambda - PB_1 \\ * & W \end{bmatrix} \succ 0, \quad (13)$$

$$W = 2\Lambda - \Lambda D_{11} - D_{11}^\top \Lambda. \quad (14)$$

For the rest of the paper, we refer to NodeRENs that comply with (13)-(14) as C-NodeRENs.

**Remark 2.** One may want to impose a certain convergence rate for a C-NodeREN. For this purpose, (13) can be modified as follows:

$$\begin{bmatrix} -A^\top P - PA - 2\tilde{\omega}P & -C_1^\top \Lambda - PB_1 \\ * & W \end{bmatrix} \succ 0,$$

where the term  $-2\tilde{\omega}P$  enforces that (3) holds with  $c = \tilde{\omega}$ . This can be directly seen by following the proof of Theorem 1.

Next, we characterize NodeRENs that comply with incremental IQCs.

**Theorem 2.** A NodeREN (8)-(9) satisfies the incremental IQC described by the triple  $(Q, S, R)$  fulfilling (7), if there exists a matrix  $P \succ 0$  and a diagonal matrix  $\Lambda \succ 0$  such that

$$\begin{bmatrix} -A^\top P - PA & -C_1^\top \Lambda - PB_1 & -PB_2 + C_2^\top S^\top \\ * & W & -\Lambda D_{12} + D_{21}^\top S^\top \\ * & * & \mathcal{R} \end{bmatrix} + \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ D_{22}^\top \end{bmatrix} Q \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ D_{22}^\top \end{bmatrix}^\top \succ 0, \quad (15)$$

with  $W$  given by (14) and

$$\mathcal{R} = R + SD_{22} + D_{22}^\top S^\top. \quad (16)$$

For the rest of the paper, we refer to NodeRENs that comply with (15)-(16) as IQC-NodeRENs.

**Remark 3.** Note that all IQC-NodeRENs are also contracting for any choice of  $(Q, S, R)$  such that (7) holds. Indeed, if there exists  $P \succ 0$  and a diagonal matrix  $\Lambda \succ 0$  such that (15) holds, then (13) also holds because  $Q \preceq 0$  and the left-hand-side of (13) is a principal minor of that of (15).

It should be noted that solving the minimization problem (2) using the above parametrizations of C-NodeRENs or IQC-NodeRENs requires solving a semidefinite program (SDP) multiple times during the optimization process. This can be computationally intractable for DNNs with several parameters. In the following section, we propose a parametrization of a rich class of C-NodeRENs and IQC-NodeRENs that circumvents this problem and allows for unconstrained optimization. It is worth noting that this approach is similar to [6], but the techniques differ due to the continuous-time nature of the proposed NodeRENs.

#### B. Direct Parametrization of NodeRENs

Our goal is to define new free parameters  $\theta_C \in \mathbb{R}^{n_C}$  and  $\theta_{IQC} \in \mathbb{R}^{n_{IQC}}$  that can be mapped onto the parameters  $\theta$  of the NodeREN model, given by equations (8)-(9), and such that either (13) or (15) are verified. First, we focus on constructing C-NodeRENs from the parameters  $\theta_C \in \mathbb{R}^{n_C}$  given by

$$\theta_C = \{X, B_2, C_2, D_{12}, D_{21}, D_{22}, \tilde{b}, U, Y_1, X_P\}, \quad (17)$$

where  $X \in \mathbb{R}^{(n+q) \times (n+q)}$ ,  $B_2 \in \mathbb{R}^{n \times m}$ ,  $C_2 \in \mathbb{R}^{p \times n}$ ,  $D_{12} \in \mathbb{R}^{q \times m}$ ,  $D_{21} \in \mathbb{R}^{p \times q}$ ,  $D_{22} \in \mathbb{R}^{p \times m}$ ,  $\tilde{b} \in \mathbb{R}^{(n+q+p)}$ ,  $U \in \mathbb{R}^{n \times q}$ ,  $Y_1 \in \mathbb{R}^{n \times n}$  and  $X_P \in \mathbb{R}^{n \times n}$ .

**Theorem 3.** For any  $\theta_C \in \mathbb{R}^{n_C}$  defined in (17), and  $\epsilon, \epsilon_P > 0$  the two following statements hold.

- 1) There are matrices  $(Y, W, Z, P)$  of appropriate dimensions such that

$$\begin{bmatrix} -Y^\top - Y & -U - Z \\ * & W \end{bmatrix} = X^\top X + \epsilon I, \quad (18)$$

$$P = X_P^\top X_P + \epsilon_P I. \quad (19)$$

- 2) There are matrices  $\{\tilde{A}, \tilde{b}\}$  defined in terms of  $\theta_C$  and the matrices  $(Y, W, Z, P)$  defined in point 1, such that the corresponding NodeREN (8)-(9) is contracting.

Moreover, matrices  $(Y, W, Z, P)$  and  $\{\tilde{A}, \tilde{b}\}$  can be computed as described in the proof of the Theorem.

In conclusion, an unconstrained parametrization of a class of C-NodeRENs is obtained by first freely choosing  $\theta_C$  defined in (17), and then recovering the matrix  $\tilde{A}$  following the steps reported in the proof (see [13]).

Next, we focus on NodeRENs that satisfy IQC constraints by design, i.e. for any choice of parameters  $\theta_{IQC} \in \mathbb{R}^{n_{IQC}}$

given by

$$\theta_{IQC} = \{X_R, B_2, C_2, D_{21}, \tilde{b}, X_3, T, U, Y_1, X_P\}, \quad (20)$$

where  $X_R \in \mathbb{R}^{(n+q) \times (n+q)}$ ,  $B_2 \in \mathbb{R}^{n \times m}$ ,  $C_2 \in \mathbb{R}^{p \times n}$ ,  $D_{21} \in \mathbb{R}^{p \times q}$ ,  $\tilde{b} \in \mathbb{R}^{(n+p+q)}$ ,  $X_3 \in \mathbb{R}^{s \times s}$ ,  $T \in \mathbb{R}^{q \times m}$ ,  $U \in \mathbb{R}^{n \times q}$ ,  $Y_1 \in \mathbb{R}^{n \times n}$  and  $X_P \in \mathbb{R}^{n \times n}$ . In the next theorem, we provide a procedure to construct an IQC-NodeREN for any choice of  $\theta_{IQC}$ .

**Theorem 4.** Let  $(Q, S, R)$  be such that (7) holds. Assume also that there exists  $\delta > 0$  satisfying  $R - S(Q - \delta I)^{-1}S^\top \succ 0$ . For any  $\theta_{IQC} \in \mathbb{R}^{n_{IQC}}$  defined in (20), and  $\epsilon, \epsilon_P > 0$  the two following statements hold.

- 1) There are matrices  $(Y, W, Z, P, D_{22})$  of appropriate dimensions such that

$$\tilde{\mathcal{R}} = R + SD_{22} + D_{22}^\top S^\top + D_{22}^\top QD_{22} \succ 0, \quad (21)$$

$$\begin{bmatrix} -Y^\top - Y & -U - Z \\ * & W \end{bmatrix} - \Psi = X_R^\top X_R + \epsilon I, \quad (22)$$

where  $P$  is constructed as in (19), and

$$\Psi = \begin{bmatrix} V \\ \tilde{T} \end{bmatrix} \tilde{\mathcal{R}}^{-1} [*]^\top - \begin{bmatrix} C_2^\top \\ D_{21}^\top \end{bmatrix} Q [*]^\top, \quad (23)$$

$$\tilde{T} = -T + D_{21}^\top S^\top + D_{21}^\top QD_{22}, \quad (24)$$

$$V = -PB_2 + C_2^\top S^\top + C_2^\top QD_{22}. \quad (25)$$

- 2) There are matrices  $\{\tilde{A}, \tilde{b}\}$  defined in terms of  $\theta_{IQC}$  and the matrices  $(Y, Z, W, P, D_{22})$  defined in point 1, such that the corresponding NodeREN (8)-(9) satisfies the incremental IQCs parametrized by  $(Q, S, R)$ .

Moreover, matrices  $(Y, W, Z, P)$  and  $\{\tilde{A}, \tilde{b}\}$  can be computed as described in the proof of the Theorem.

The proof of Theorem 4 shares similarities with the one of Proposition 2 in [6] and it is reported in [13]. Note that the assumption that there exists a  $\delta$  such that  $R - S(Q - \delta I)^{-1}S^\top \succ 0$  is not restrictive. Indeed, for all the most relevant cases, reported in Table I, finding an appropriate value of  $\delta$  is straightforward. In conclusion, an unconstrained parametrization of IQC-NodeRENs is obtained by first freely choosing the parameters  $\theta_{IQC} \in \mathbb{R}^{n_{IQC}}$ , then constructing  $D_{22}, \tilde{A}$  following the steps reported in the full proof.

#### IV. SIMULATIONS & RESULTS

In this section, we showcase the application of NodeRENs for identifying stable nonlinear systems. We begin by assessing the performance of NodeRENs when using various integration methods during training. Next, we highlight the advantages of contractivity-by-design by training a general non-contractive NodeREN (10)-(11), which fails to learn stable behavior for the same task. Finally, we demonstrate the ability of NodeRENs to learn from irregularly sampled time-series data while maintaining the guarantees by design. Our code is accessible at <https://github.com/DecodePFL/NodeREN>.

We remark that NodeREN trajectories are time-reversible, as established in Lemma 1, which implies that they cannot

intersect for distinct initial conditions. To enhance model expressiveness, we employ augmented state vectors, initializing additional scalar states to zero. This technique, known as feature augmentation, was introduced in [21] for general Neural ODEs.

##### A. Continuous-Time System Identification

Here, we consider the system identification of a black-box system. For this experiment, we assume that the unknown system dynamics are those of a nonlinear pendulum

$$\ell \ddot{\alpha}(t) + \beta \dot{\alpha}(t) + g \sin \alpha(t) = 0, \quad (26)$$

where  $\alpha(t)$  is the angle position of the pendulum at time  $t$  with respect to the vertical axis,  $\beta > 0$  is the viscous damping coefficient,  $g$  is the gravitational acceleration and  $\ell > 0$  is the length of the pendulum.

We consider the scenario where the system dynamics (26) to identify are completely unknown and the only prior knowledge we have is that the system is stable around the origin. Hence, we choose to train a C-NodeREN, which is guaranteed to be contracting (and so, stable around the origin if  $\tilde{b}$  is null) for any choice of the trainable parameters  $\theta_C$  — i.e. even if we stop the optimization prematurely.

The training data are given by noisy measurements of  $\alpha(t)$  and  $\dot{\alpha}(t)$  across  $N \in \mathbb{N}$  different experiments performed in the time interval  $[0, T_{end}]$ . For each experiment, the system (26) starts from a different and known initial condition  $\alpha_i(0)$  and  $\dot{\alpha}_i(0)$ . Due to the possibility of random time delays during data acquisition, we assume that trajectory measurements are taken at random time instants across  $[0, T_{end}]$  for each experiment. We obtain the parameters  $\theta_C$  minimizing the mean squared error between the C-NodeREN predictions and the simulated measurements. For more details, please refer to [13]. In Figure 1, we report the resulting test losses for different methods, along with their Number of Function Evaluations (NFE) used for the prediction. It is important to emphasize how for fixed-step methods (e.g., `euler`, `rk4`), the chosen number of steps may be insufficient to fully capture the dynamics of the trained model, which can result in numerical instability. In contrast, adaptive methods (e.g., `dopri5`) dynamically adjust the step size to ensure bounded integration errors, hence promoting numerical stability in the trajectory simulations. Such freedom enables us to either control the NFEs with a fixed step approach, or to achieve more accurate simulations with an adaptive step size method. As anticipated in Remark 1, it would be interesting to endow C-NodeRENs with specific integration methods that preserve contractivity, e.g. [19].

Then, for the same identification task, we have trained a general NodeREN (10)-(11) (that we denote as G-NodeREN), with trainable parameters  $\theta = \{\tilde{A}, \tilde{b}\}$ . Given the higher flexibility of G-NodeRENs with respect to C-NodeRENs, one could expect a better performance in the time window  $[0, T_{end}]$  considered for the training. However, a trained G-NodeREN can exhibit unstable dynamics when simulating for longer horizons; this phenomenon is illustrated in Figure 2. Instead, C-NodeRENs only limit the search to

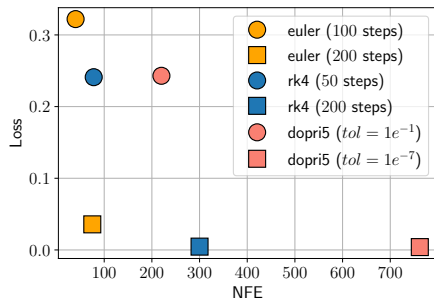


Fig. 1: Comparison between the number of function evaluations (NFE) and loss on testing dataset for three integration methods: forward Euler (`euler`), Runge-Kutta IV (`rk4`), and Dormand-Prince-Shampine (`dopri5`).

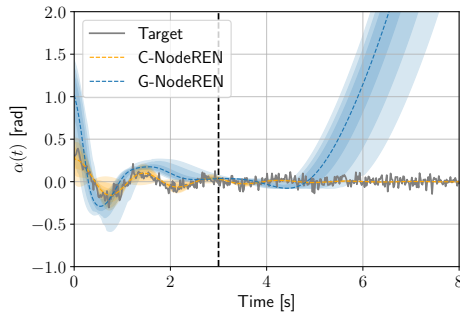


Fig. 2: Noisy test trajectory of the angular position  $\alpha(t)$  of the real system (in gray). Predicted trajectories of the trained C-NodeREN (in orange) and G-NodeREN (in blue). Shaded areas represent tubes of additional trajectories resulting from perturbations on the initial state of the systems. The black dashed line indicates the end of the training horizon.

stable models, resulting in better identification performance for the considered example. In Figure 2, we also display trajectory tubes from different perturbed initial conditions. Note that the diameter of the tube associated with the C-NodeREN decreases over time, showcasing its contractivity.

### B. Irregularly sampled-data

To show the robustness of C-NodeRENs concerning the irregular sample data, we compare the test losses of 10 C-NodeRENs trained on 10 different training datasets having the same initial conditions, but different sampling times. We observe that, despite using differently sampled data, all the models have similar test losses: they all lay in the interval  $[3.7 \times 10^{-4}; 9.1 \times 10^{-3}]$ , depending on how well the samples were distributed.

## V. CONCLUSIONS

In this work, we have established a class of Neural ODEs that generalize RENs for continuous-time scenarios. Specifically, the proposed models guarantee relevant continuous-time system-theoretic properties such as contractivity and dissipativity by design. The resulting DNNs can be trained using unconstrained optimization — which enables the use of large networks — and the resulting architectures are flexible

in the choice of an ODE integration scheme that is suitable for the learning problem at hand. We have showcased the performance of NodeRENs through the task of identifying the behavior of a nonlinear pendulum, even with irregularly sampled data.

Future research should focus on integration schemes preserving contractivity and IQC properties, exploring distributed NodeREN architectures, and analyzing their generalization capabilities.

## REFERENCES

- [1] I. H. Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN computer science*, vol. 2, no. 3, p. 160, 2021.
- [2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *International Conference on Learning Representations*, 2014.
- [3] E. Haber and L. Ruthotto, “Stable architectures for deep neural networks,” *Inverse Problems*, vol. 34, p. 014004, dec 2017.
- [4] M. Zakwan, L. Xu, and G. Ferrari-Trecate, “Robust classification using contractive Hamiltonian neural ODEs,” *IEEE Control Systems Letters*, vol. 7, pp. 145–150, 2023.
- [5] K.-K. Kim, E. R. Patrón, and R. D. Braatz, “Standard representation and unified stability analysis for dynamic artificial neural network models,” *Neural Networks*, vol. 98, pp. 251–262, 2018.
- [6] M. Revay, R. Wang, and I. R. Manchester, “Recurrent equilibrium networks: Flexible dynamic models with guaranteed stability and robustness,” *IEEE Transactions on Automatic Control*, 2023.
- [7] R. Wang and I. R. Manchester, “Youla-REN: Learning nonlinear feedback policies with robust stability guarantees,” *2022 American Control Conference (ACC)*, pp. 2116–2123, 2021.
- [8] K. J. Åström, P. Hagander, and J. Sternby, “Zeros of sampled systems,” *Automatica*, vol. 20, no. 1, pp. 31–38, 1984.
- [9] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” *Advances in Neural Information Processing Systems*, 2018.
- [10] P. Kidger, J. Morrill, J. Foster, and T. Lyons, “Neural controlled differential equations for irregular time series,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6696–6707, 2020.
- [11] C. L. Galimberti, L. Furiieri, L. Xu, and G. Ferrari-Trecate, “Hamiltonian deep neural networks guaranteeing non-vanishing gradients by design,” *IEEE Transactions on Automatic Control*, 2023.
- [12] S. Massaroli, M. Poli, M. Bin, J. Park, A. Yamashita, and H. Asama, “Stable neural flows,” *arXiv preprint arXiv:2003.08063*, 2020.
- [13] D. Martinelli, C. L. Galimberti, I. R. Manchester, L. Furiieri, and G. Ferrari-Trecate, “Unconstrained parametrization of dissipative and contracting neural ordinary differential equations,” *arXiv preprint arXiv:2304.02976*, 2023.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [15] E. Hairer, G. Wanner, and S. P. Nørsett, *Solving Ordinary Differential Equations I*. Springer Berlin Heidelberg, 1993.
- [16] Q. Kang, Y. Song, Q. Ding, and W. P. Tay, “Stable neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 14925–14937, 2021.
- [17] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.
- [18] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, “Training robust neural networks using Lipschitz bounds,” *IEEE Control Systems Letters*, vol. 6, pp. 121–126, 2021.
- [19] I. R. Manchester, “Contracting nonlinear observers: Convex optimization and learning from data,” in *2018 Annual American Control Conference (ACC)*, pp. 1873–1880, 2018.
- [20] J. Lygeros and F. Ramponi, “Lecture notes on linear system theory,” *Automatic Control Laboratory, ETH Zurich*, 2015.
- [21] E. Dupont, A. Doucet, and Y. W. Teh, “Augmented neural ODEs,” *Advances in neural information processing systems*, vol. 32, 2019.