POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Contracting and Robust Deep Neural Network in Continuous Time

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Daniele Martinelli

Advisor: Prof. Riccardo Scattolini

Co-advisors: Prof. Giancarlo Ferrari Trecate, Luca Furieri, Clara Galimberti

Academic year: 2021-2022

---

1. Introduction

In this thesis, we present a new class of deep neural networks in continuous time that can be used in different learning tasks, from classification problems to control applications such as continuous-time system identification and optimal control. The name of this architecture is: Recurrent Equilibrium Network Ordinary Differential Equations (REN-ODEs). This new class of neural networks consists of nonlinear dynamical systems that assures contractivity (a powerful form of stability) *by design* and can also guarantee incremental *integral quadratic constraints* (IQCs). IQCs are used to enforce properties of incremental dissipativity and passivity, as well as Lipschitz bounds. These properties provide *robustness* to the model. With this term, we mean a mitigation of the sensitivity of the system's outputs with respect to small perturbations in the inputs. This feature is important in applications in which signals are affected by noise (e.g., system identification from real acquired data). Being contractive and robust *by design* means that the N parameters, characterizing a REN-ODE, are unconstrained. This property makes possible, during the learning phase, to use *unconstrained* iterative first-order optimization methods such as *gradient descent* (and its variations). The structure of the class is inspired by the Recurrent Equilibrium Networks (RENs) [1], which, however, are formulated in discrete time. Moreover, the REN-ODE's architecture belongs to the family of Neural Ordinary Differential Equations (Neural-ODEs) [2] . As a result, REN-ODEs inherit all the advantages of Neural-ODEs, including the possibility to use modern and sophisticated ODE solvers for the evaluation of the model's trajectories. Furthermore, latest ODE solvers can provide high level of precision and adapt the evaluation strategy on the fly to achieve the requested level of accuracy. In this work, the properties of contractivity and robustness are validated on a nonlinear system identification problem and an optimal control task. Moreover, we evaluate the performance of the REN-ODEs over benchmark binary classification problems.

2. Preliminary Knowledge

Non-linear system analysis is still an open and hot topic due to the complexity of the subject. Concepts as *contractivity*, *dissipativity* and *passivity* are frequently used in

this field. In this section, these definitions are provided.

Notation With $x^a(t)$, where $x, a \in \mathbb{R}^n$, we denote the state x at time t starting from the initial condition a at time t_0. A function is $\mathcal{C}^1$ if continuous and differentiable. The finite 2-norm of the signal y from time t_0 up to time T is denoted as $\|y\|_T$, i.e.,

$$\|y\|_T = \Big(\int_{t_0}^{T} |y(\tau)|^2 d\tau\Big)^{1/2}.$$

With $X \geq 0$ and $X \leq 0$ we denote that the generic square matrix X is semi-positive definite and semi-negative definite, respectively. Finally, $\mathbb{1}(t)$ represents the unit-step signal.

2.1. Contractivity

We consider a general deterministic dynamical system Σ of the form:

$$\Sigma = \begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = g(x(t), u(t)) \end{cases} \tag{1}$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $y \in \mathcal{Y} \subseteq \mathbb{R}^p$, $u \in \mathcal{U} \subseteq \mathbb{R}^m$ are respectively the system state, output and input. Furthermore, $f : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ and $g : \mathcal{X} \times \mathcal{U} \to \mathcal{Y}$ are the state evolution and output functions. The functions f and g are $\mathcal{C}^1$.

Definition 2.1 (Contracting System)**.** A system Σ in the form (1) is said to be *contracting* if for any two initial conditions a,$b \in \mathcal{X}$, the state function $x^a(t)$ and $x^b(t)$ with the same input function $u(t)$ satisfy:

$$\|x^a(t) - x^b(t)\| \leq \kappa e^{-c(t-t_0)} \|a - b\|, \quad \forall t \geq t_0 \tag{2}$$

for some $c > 0$ and $\kappa > 0$.

The Definition 2.1 can be interpreted as follows: a contractive model is a model that "forgets" the initial condition with a certain rate c. This property can be useful especially in applications such as system identification or state estimation (designing of a state observer), in which the initial state can be affected by uncertainty.

2.2. Dissipativity

We define *supply rate* a function $s(u(t), y(t))$:

$$s : \mathcal{U} \times \mathcal{Y} \to \mathbb{R}. \tag{3}$$

Dissipativity, in many physical systems, may be roughly interpreted as the way the system exchanges its internal energy with the external through the inputs and outputs. Thus, it is intuitive how the concept of dissipativity is important in control theory: under certain conditions of controllability and reachability, the way the system stores and expels energy, based on inputs and outputs, can provide information about the overall stability of the process. At this point, we introduce the definition of a dissipative system.

Definition 2.2 (Generally Dissipative System)**.** Given a *supply rate* $s()$, a system Σ in the form (1) is said to be *dissipative* with respect to $s()$ if there exists a function $\mathcal{S} : \mathcal{X} \to \mathbb{R}^+$, called *storage function*, such that for any initial condition $x(t_0) \in \mathcal{X}$ at any time t_0, and for any input $u(\cdot) \in \mathcal{U}$ and the following inequality holds:

$$\mathcal{S}(x(t_1)) \leq \mathcal{S}(x(t_0)) + \int_{t_0}^{t_1} s(u(t), y(t)) dt, \quad \forall t_1 \geq t_0 \tag{4}$$

where the integral is assumed to be well defined for all allowed $u(\cdot) \in \mathcal{U}$ and $y(\cdot) \in \mathcal{Y}$.

If the function $\mathcal{S}(x(t))$ is differentiable, (4) can be rewritten as:

$$\frac{d}{dt}\Big(\mathcal{S}(x(t))\Big) \leq s\big(u(t), y(t)\big). \tag{5}$$

This is called *differentiated dissipation inequality*.

2.3. Incremental Dissipativity

At this point, it is possible to provide the notion of *incremental* dissipativity, i.e., the study of the energy flow between any two trajectories of the system, that may variate due to possible different initial conditions or sequence of inputs. We can denote with:

$$\Delta y(t) = y(t) - \tilde{y}(t)\,, \quad \Delta u(t) = u(t) - \tilde{u}(t)\,, \tag{6}$$

$$\Delta x(t) = x(t) - \tilde{x}(t)\,, \quad \Delta v(t) = v(t) - \tilde{v}(t)\,, \tag{7}$$

the finite differences between the two possible trajectories $(x, u, y), (\tilde{x}, \tilde{u}, \tilde{y})$ of the system Σ.

Definition 2.3 (Incrementally Dissipative System)**.** Given a *supply rate* $s()$, a system Σ in the form (1) is said to be *incrementally dissipative* with respect to $s()$ if there exists a storage function $\mathcal{S} : \mathcal{X} \to \mathbb{R}^+$ such that for any two possible trajectories $(x, u, y), (\tilde{x}, \tilde{u}, \tilde{y})$ and:

$$\mathcal{S}(\Delta x(t_1)) \leq \mathcal{S}(\Delta x(t_0)) + \int_{t_0}^{t_1} s(\Delta u(t), \Delta y(t)) dt\,, \quad \forall t_1 \geq t_0 \tag{8}$$

where the integral is assumed to be well defined for all allowed $u(\cdot) \in \mathcal{U}$ and $y(\cdot) \in \mathcal{Y}$.

2.4. Passivity

As an assumption, a necessary condition for a system to be passive is that the input and output must have the same dimensions, i.e., $m \equiv p$.

Definition 2.4 (Passive System)**.** A system Σ in the form (1) is said to be *passive* if it is dissipative with respect to the supply rate $s(u, y) = u^\top y$.

Additionally, if Σ is dissipative to some particular classes of $s()$, then Σ is:

- *input strictly passive* if

$$s(u, y) = u^\top y - \nu\|u\|^2 \,,\ \nu > 0 \tag{9}$$

- *output strictly passive* if

$$s(u, y) = u^\top y - \varepsilon\|y\|^2 \,,\ \varepsilon > 0 \tag{10}$$

2.5. Integral Quadratic Constraint (IQC)

Incremental integral quadratic constraints (IQCs) can be seen as a definition of incremental dissipativity where the supply rate takes this parametrized form:

$$s(\Delta u, \Delta y) = \begin{bmatrix}\Delta y(t)\\ \Delta u(t)\end{bmatrix}^\top \begin{bmatrix}Q & S^\top\\ S & R\end{bmatrix} \begin{bmatrix}\Delta y(t)\\ \Delta u(t)\end{bmatrix}, \tag{11}$$

with $Q \in \mathbb{R}^{p\times p}$, $R \in \mathbb{R}^{m\times m}$ and $S \in \mathbb{R}^{m\times p}$.

Definition 2.5 (Incremental Integral Quadratic Constraint (IQC))**.** A system Σ in the form (1) is said to satisfy the *incremental integral quadratic constraints* (IQCs) defined by the matrices (Q, S, R) where $0 \geq Q$ and $R = R^T$ if it is incrementally dissipative with respect to the supply rate in (11).

One of the main interesting applications of incremental IQCs is the possibility to verify properties of the system (such as dissipativity and passivity) just choosing proper values of the fixed matrices (Q, S, R). Indeed, taking:

- $Q = -\frac{1}{\gamma}I, R = \gamma I, S = 0 \rightarrow$ the model satisfies an ℓ_2 Lipschitz bound of γ:

$$\|y - \bar{y}\|_{t_1} \leq \gamma\|u - \bar{u}\|_{t_1} \,, \quad \forall t_1 \geq t_0. \tag{12}$$

- $Q = 0, R = -2\nu I, S = I, \nu \geq 0 \rightarrow$ the system is incrementally input passive.
- $Q = -2\varepsilon I, R = 0, S = I, \varepsilon \geq 0 \rightarrow$ the system is incrementally output passive.

3. REN-ODE Model

Our contribution in this work is the introduction of a new deep neural network model: REN-ODE. REN-ODE shares similar structure to the original REN [1]. The system model is the following:

$$\begin{bmatrix}\dot{x}_t\\ v_t\\ y_t\end{bmatrix} = \begin{bmatrix}A & B_1 & B_2\\ C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\end{bmatrix} \begin{bmatrix}x_t\\ w_t\\ u_t\end{bmatrix} + \begin{bmatrix}b_x\\ b_v\\ b_y\end{bmatrix} \tag{13}$$

$$w_t = \sigma(v_t) \tag{14}$$

where $x_t \in \mathbb{R}^n$, $v_t \in \mathbb{R}^q$, $w_t \in \mathbb{R}^q$, $u_t \in \mathbb{R}^m$ and $y_t \in \mathbb{R}^p$ are respectively the state, the nonlinear output, the nonlinear input, the exogenous input and the linear output. With $\sigma(\cdot)$ we denote a nonlinear function applied elementwise called "activation function". Consider the *finite* difference between two possible trajectories (x, w, v, u) and $(\bar{x}, \bar{w}, \bar{v}, \bar{u})$ having different initial conditions and input functions. At this point, the *incremental form* of the system is:

$$\begin{bmatrix}\Delta\dot{x}_t\\ \Delta v_t\\ \Delta y_t\end{bmatrix} = \begin{bmatrix}A & B_1 & B_2\\ C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\end{bmatrix} \begin{bmatrix}\Delta x_t\\ \Delta w_t\\ \Delta u_t\end{bmatrix}, \tag{15}$$

$$\Delta w_t = \sigma(v_t + \Delta v_t) - \sigma(v_t). \tag{16}$$

In order to achieve properties of contractivity and robustness later, some assumptions must be introduced.

Assumption 1 (Rate-Limited σ)**.** The activation function $\sigma(\cdot)$ is piece-wise differentiable and slope-restricted in $[0, 1]$, i.e.,

$$0 \leq \frac{\sigma(y) - \sigma(x)}{y - x} \leq 1 \,, \quad \forall x, y \in \mathbb{R} \,, \quad x \neq y. \tag{17}$$

The inequality (17) can be also rewritten for each j^{th} channel as a conic combination with multipliers $\xi_j > 0$, resulting in an *incremental* IQC:

$$\Gamma_t = \begin{bmatrix}\Delta v_t\\ \Delta w_t\end{bmatrix}^\top \begin{bmatrix}0 & \Lambda\\ \Lambda & -2\Lambda\end{bmatrix} \begin{bmatrix}\Delta v_t\\ \Delta w_t\end{bmatrix} \geq 0, \ \forall t \in \mathbb{R}. \tag{18}$$

where $\Lambda = \text{diag}(\xi_1, \ldots, \xi_q)$. Δv_t and Δw_t represent the difference between the two trajectories having (v_t, w_t) and $(\bar{v}_t, \bar{w}_t)$.

Well-Posedness An important point is to show that the REN-ODE is *well-posed*, i.e., given any particular input $(\bar{x}_k, \bar{u}_k, \bar{b}_v)$, a unique solution $\bar{w}_k$ exists. In formula:

$$\begin{aligned}&\forall \bar{x}_k, \bar{u}_k, \bar{b}_v \quad \text{then} \quad \exists!\ \bar{w}_k :\\ &\bar{w}_k = \sigma\big(D_{11}\bar{w}_k + C_1\bar{x}_k + D_{12}\bar{u}_k + \bar{b}_v\big).\end{aligned} \tag{19}$$

It has been shown in [3] that if there exists a positive definite diagonal matrix Λ such that:

$$2\Lambda - \Lambda D_{11} - D_{11}^\top\Lambda > 0, \tag{20}$$

then the problem in (14) is well-posed.

Acyclicity A subset of REN-ODEs are the *acyclic* REN-ODEs. These models are characterized by the weight matrix D_{11} constrained to be strictly lower triangular; it means that the i^{th} channel of $v(k)$ with $i \in [1, q]$ will depend only by the channels from $\{1, \dots (i-1)\}$. Assuming an acyclic model allows to simplify the numerical calculation of the trajectories of the network.

4. Contracting and Robust REN-ODEs

The main goal of this section is to obtain two unconstrained parametrizations (called *direct parametrizations*) of REN-ODEs that are able to guarantee respectively contractivity and robustness by design. Firstly, the necessary conditions for contractivity and robustness of REN-ODEs are provided here.

Theorem 4.1 (Contractive REN-ODE). *A REN-ODE in the form* (13)-(14) *is contracting if there exists a matrix* $P > 0$ *and a diagonal matrix* $\Lambda > 0$ *such that*

$$\begin{bmatrix} -A^\top P - PA & -C_1^\top \Lambda - PB_1 \\ -\Lambda C_1 - B_1^\top P & W \end{bmatrix} > 0, \tag{21}$$

with

$$W = 2\Lambda - \Lambda D_{11} - D_{11}^\top \Lambda. \tag{22}$$

The proof is reported in Appendix A.

Theorem 4.2 (Robust REN-ODE). *A REN-ODE in the form* (13)-(14) *is well-posed and satisfies the incremental IQC described by* (Q, S, R) *if there exists a matrix* $P > 0$ *and a diagonal matrix* $\Lambda > 0$ *such that*

$$\begin{bmatrix} -A^\top P - PA & -PB_1 - C_1^\top \Lambda & -PB_2 + C_2^\top S^\top \\ -B_1^\top P - \Lambda C_1 & W & D_{21}^\top S^\top - \Lambda D_{12} \\ -B_2^\top P + SC_2 & SD_{21} - D_{12}^\top \Lambda & R + SD_{22} + D_{22}^\top S^\top \end{bmatrix} + \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ D_{22}^\top \end{bmatrix} Q \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ D_{22}^\top \end{bmatrix}^\top > 0, \tag{23}$$

with W *given by* (22).

The proof is reported in Appendix B.

4.1. Convex Parametrization

It is possible to notice that the matrix inequalities in (21), (23) are not convex in their parameters (e.g., the presence of the element $A^\top P$). This detail would not make achievable a direct parametrization later. For this reason, it is important to rewrite the matrix inequalities in a convex form with respect to their parameters.

Starting with (21), it is possible to cast it, under a suitable reformulation, as a linear one. To do so, we define the following matrices:

$$U = C_1^\top \Lambda\,, \;\; Y = PA\,, \;\; Z = PB_1 \tag{24}$$

with $U \in \mathbb{R}^{n \times q}, Y \in \mathbb{R}^{n \times n}, Z \in \mathbb{R}^{n \times q}$. Then (21) becomes:

$$\begin{bmatrix} -Y^\top - Y & -U - Z \\ -U^\top - Z^\top & W \end{bmatrix} > 0 \tag{25}$$

Note that, now, the inequality (25) is linear and convex, with the independent variables (U, W, Y, Z) as free/design variables. We also want the matrix inequality (23) to be convex in its parameters; to do so, we can define, the matrices:

$$\tilde{\mathcal{V}} = -PB_2 + C_2^\top S^\top + C_2^\top Q D_{22}\,, \tag{26}$$

$$\tilde{T} = -\Lambda D_{12} + D_{21}^\top S^\top + D_{21}^\top Q D_{22}\,, \tag{27}$$

$$\mathcal{R} = R + SD_{22} + D_{22}^\top S^\top + D_{22}^\top Q D_{22}. \tag{28}$$

with $\tilde{\mathcal{V}} \in \mathbb{R}^{n \times m}, \tilde{T} \in \mathbb{R}^{q \times m}, \mathcal{R} \in \mathbb{R}^{m \times m}$. It is easy to check that, using (24), (26)-(28), (23) can be rearranged as:

$$\begin{bmatrix} -Y^\top - Y & -U - Z & \tilde{\mathcal{V}} \\ -U^\top - Z^\top & W & \tilde{T} \\ \tilde{\mathcal{V}}^\top & \tilde{T}^\top & \mathcal{R} \end{bmatrix} + \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ 0 \end{bmatrix} Q \begin{bmatrix} C_2^\top \\ D_{21}^\top \\ 0 \end{bmatrix}^\top > 0. \tag{29}$$

At this point, using the *Schur Complement*, it is possible to rewrite the inequality (29) as:

$$\mathcal{R} > 0, \tag{30}$$

$$\begin{bmatrix} -Y^\top - Y & -U - Z \\ -U^\top - Z^\top & W \end{bmatrix} - \begin{bmatrix} \tilde{\mathcal{V}} \\ \tilde{T} \end{bmatrix} \mathcal{R}^{-1} \begin{bmatrix} \tilde{\mathcal{V}} \\ \tilde{T} \end{bmatrix}^\top + \begin{bmatrix} C_2^\top \\ D_{21}^\top \end{bmatrix} Q \begin{bmatrix} C_2^\top \\ D_{21}^\top \end{bmatrix}^\top > 0. \tag{31}$$

However, if S is chosen as a zero matrix, then, given a fixed value of $\mathcal{R}$, solving (28) in D_{22} may return a complex matrix as solution, not acceptable for real signals. For this reason D_{22} must be chosen in such a way that, given any S and any $\mathcal{R}$, the (28) is valid, with D_{22} real. In Appendix C, it is reported how to construct D_{22}.

4.2. Direct Parametrization

Once the matrix inequalities (21), (23) have been cast in a convex form in their parameters (i.e., (25), (29)), then direct parametrization can be finally retrieved.

Convex Direct Parametrization The main principle is to parameterize the leftside matrix from (21), as $X^\top X + \epsilon I$, with $X \in \mathbb{R}^{(n+q)\times(n+q)}$ a free matrix variable and $\epsilon > 0$

and then retrieve all the parameters of the system. Denote $H \in \mathbb{R}^{(n+q)\times(n+q)}$ the matrix:

$$H = \begin{bmatrix} -Y^\top - Y & -U - Z \\ -U^\top - Z^\top & W \end{bmatrix} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix}. \tag{32}$$

Then parameterizing it as $H = X^\top X + \epsilon I$, makes (21) always true by construction. At this point, from (32), it is easy to see that:

$$-Y^\top - Y = H_{11}\,, \quad -U - Z = H_{12}. \tag{33}$$

From (33), Y can be parameterized as:

$$Y = -\frac{1}{2}(H_{11} + Y_1 - Y_1^\top). \tag{34}$$

where $Y_1 \in \mathbb{R}^{n\times n}$ is a free matrix that allows to obtain also non-symmetric Y. Considering (33), it is possible to keep one of the matrices (U, Z) free (e.g., U), and then obtain the remaining one by the relation (33). In case U is chosen freely, then:

$$Z = -H_{12} - U. \tag{35}$$

If the system is *acyclic*, then D_{11} must be strictly lower triangular, thus from H_{22}:

$$H_{22} \;=\; W \;=\; 2\Lambda - D_{11} - D_{11}^\top, \tag{36}$$

it is possible to obtain Λ from $1/2$ of the main diagonal of W and D_{11} from the strict lower triangular part of W. It must be noted that the matrix P never appears by itself, thus it does not need to be calculated from H. However, P must be positive definite. In order to enforce this condition, P can be built as follows:

$$P = P_1 P_1^\top + \epsilon I > 0\,, \qquad \forall P_1 \in \mathbb{R}^{n\times n}. \tag{37}$$

where P_1 is a generic free matrix and $\epsilon > 0$.
From (33)-(37) the matrices A, B_1, C_1 can be retrieved. The remaining parameters do not impact the contractivity of the model and, thus, they can be considered free. In conclusion, to define a CREN-ODE, first one can set the free parameters $X, B_2, C_2, D_{12}, D_{21}, D_{22}, b_x, b_v, b_y, P_1, U$ and Y_1. Then, the remaining matrices are recovered using (33)-(37). In order to update the parameters, it is possible to use the most common unconstrained optimization methods such as *gradient descent* and its variations (e.g. SGD, Adam). For a deeper review about the topic, the reader is referred to [4].

Robust Direct Parametrization In the robust direct parametrization, we can follow similar steps as in the contractive case. We denote with $\tilde{d} = n + q + m$. We call H the matrix from (31) and we parameterize it with X and ϵ ($X, H \in \mathbb{R}^{\tilde{d}\times\tilde{d}}, \epsilon > 0$):

$$\begin{aligned} H = &\begin{bmatrix} -Y^\top - Y & -U - Z \\ -U^\top - Z^\top & W \end{bmatrix} - \begin{bmatrix} \tilde{\mathcal{V}} \\ \tilde{T} \end{bmatrix} \mathcal{R}^{-1} \begin{bmatrix} \tilde{\mathcal{V}} \\ \tilde{T} \end{bmatrix}^\top \\ &+ \begin{bmatrix} C_2^\top \\ D_{21}^\top \end{bmatrix} Q \begin{bmatrix} C_2^\top \\ D_{21}^\top \end{bmatrix}^\top \qquad (38) \\ = &\, X^\top X + \epsilon I. \qquad (39) \end{aligned}$$

At this point, if D_{22} is constructed as described in (53) and assumed that $Q < 0$ and $R = R^\top$, then the inequalities to prove robustness of REN-ODE (30), (31) are verified. This means that, for any choice of the free parameters (i.e. X, B_2, C_2, T, D_{21}, X_3, Y_3, b_x, b_v, b_y, P_1, U, Y_1), then the system always satisfies the incremental IQCs described by (Q, S, R). Then, the remaining matrices are recovered using (26)-(28), (33)-(37) and (50)-(53).

5. Subclass of Neural-ODEs

The class of neural networks that we propose in this thesis (REN-ODE) belongs to the family of Neural-ODEs. Indeed, the authors presented in [2] an innovative recurrent neural network (RNN) in which the number of hidden layers is not specified; instead, a *continuous number of layers* is considered, where the derivative of the hidden state is parameterized using a neural network (Figure 1). Thanks to that, the trajectories of the system can be obtained using modern ODE solvers, that can guarantee high level of precision and adapt the evaluation strategy on the fly to achieve the requested level of accuracy. Although ODE solvers may involve complex calculations, it is possible to show that, for training, it is possible to backpropagate in a scalable way, without access to its internal operations. This allows to implement also models with a high number of parameters. For more details, the reader may refer to [2].

6. Comments on REN-ODEs

In the main thesis, it has been proven that, differently from the discrete-time REN [1], the direct parametrization for robust REN-ODEs implies also contractivity. A scheme of this is reported in Table 1. This is an important result: given a robust (and contractive) RREN-ODE, if there exists a stable equilibrium point of the system for a given in-

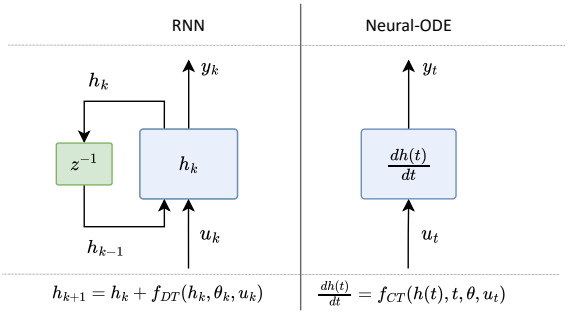


Figure 1: Representations of an hidden-layer in a generic RNN (left) and in a Neural-ODE (right). The unitary delay is represented with z^{-1}.

put, then, for any initial condition, the model will always converge exponentially to the equilibrium while keeping guarantees of *incremental* dissipativity. Additionally, this incremental form of robustness also becomes valid for the *general* form of the system, as shown in [5, Theorem 10]. This opens up to new possibilities: for example, if a passive REN-ODE is used to control a passive system (through an optimal policy) then, it can be proven that the overall negative feedback interconnection is still passive (proof in [6]). For instance, this property is considered in the optimal control study case reported in Section 7.2, where the multi-agent system to be controlled is passive.

Property	CREN	RREN	CREN-ODE	RREN-ODE
Contractivity	✓		✓	✓
Robustness		✓		✓

Table 1: Properties of the different models.

7. Simulations & Results

We used REN-ODE in different fields of application: binary classification, system identification and optimal control. Initially, simulations to validate the properties of contractivity and robustness will be reported. The implementation was carried out using Python as main programming language. The main third-party libraries used for REN-ODEs are: *PyTorch* for the neural network framework; *Torchdiffeq*, developed by the authors of the Neural-ODE architecture (it supports different integration methods such as Runge-Kutta of order 4 or 5 of Dormand-Prince-Shampine).

7.1. Validation

Firstly, we simulate a CREN-ODE starting from two different initial conditions $a, b \in \mathbb{R}^n$ randomly drawn from a normal distribution with zero mean and unitary variance. The input is an unit step-signal. In Figure 2, it is clearly visible how all the modules of the difference between the two trajectories' states are always lower (or equal) than an exponential curve $\kappa e^{-ct}|b-a|$, with $c, \kappa > 0$. Thus,the Definition 2.1 is validated. Moreover, we want to show that RREN-ODE are able to guarantee properties of ℓ_2 Lipschitz bounds. Given a value of γ (e.g., $\gamma = 1$), the simulations are carried out picking a RREN-ODE, with free parameters drawn randomly from a normal distribution with zero mean and variance $\sigma^2 = 0.01$. Then, the model has been simulated twice, using two different initial conditions $x_u(t_0), x_v(t_0)$ and two different generic inputs $u(t), v(t)$:

$$u(t) = -2e^{-0.2t}\sin\left(\frac{\pi}{2}t + \frac{\pi}{3}\right)\mathbb{1}(t) \tag{40}$$

$$v(t) = 3e^{-0.3t}\cos\left(\pi t\right)\mathbb{1}(t) \tag{41}$$

Afterwords, the finite l2-norms $\|u-v\|_t$ and $\|y_u - y_v\|_t$ are computed. In Figure 3 it is shown that (12) is guaranteed. Thus, the system is ℓ_2 Lipschitz bounded.
Subsequently, we want to validate the property of passivity for RREN-ODE. Given a prescribed form of passivity (e.g., output passivity with $\varepsilon = 10$) the system is simulated twice using the input functions (40), (41). In Figure 4 it is shown that the *differentiated* incremental form of the dissipation inequality for a strictly output passive system is satisfied, or, in other words:

$$\dot{V}_\Delta(t) < s(\Delta u_t, \Delta y_t) = \Delta u^\top \Delta y - \varepsilon\|\Delta y\|^2\,, \ \forall t. \tag{42}$$

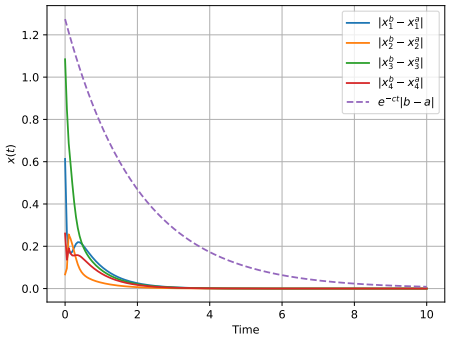


Figure 2: Plot of the module of the difference between two evolution of a model ($n = 4$) starting from two different initial conditions a, b. In violet, the curve $e^{-ct}|b-a|$.

7.2. Study Cases

In order to show the potentiality of this new neural network, REN-ODEs have been tested in different scenarios.

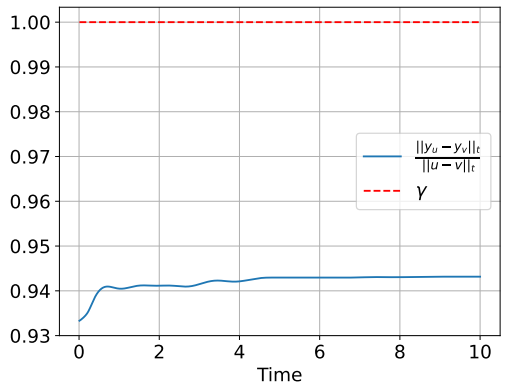


Figure 3: Plots of the normalized finite l2-norm of the difference between the outputs $y_u(t), y_v(t)$ with $\gamma = 1$.

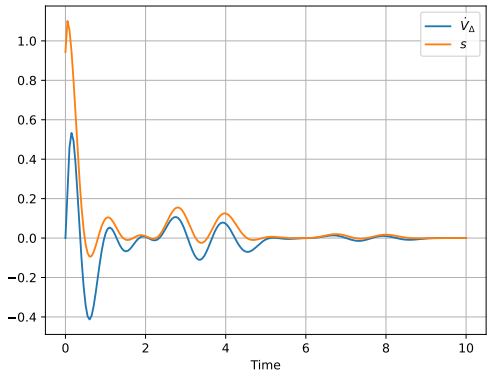
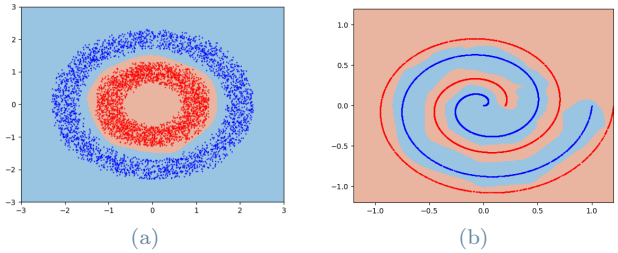


Figure 4: Plots of the storage function $\dot{V}_\Delta$ and supply rate s of strictly output passive EREN-ODE with $\varepsilon = 10$.

Binary Classification The first one is for binary classification of different benchmarks from the state-of-the-art literature. The data sets have always two classification labels $\{0, 1\}$ and two features (x_1, x_2). The classification is obtained in this way: the data-set has N points and a time window T_{end} is chosen. For any i^{th} point of the data-set (x_1^i, x_2^i), with $i \in \{1, N\}$, the sample is used as initial condition for the evolution of the REN-ODE, done up to time T_{end}; then the output $y(T_{end})$ is retrieved and passed through a Sigmoid layer returning a (normalized) value between 0 and 1. Finally, comparing this result with a 0.5 threshold, the binary classification p is obtained. We used for this task the Binary Classification Entropy (BCE) as loss function. The model was able to achieve 100% of accuracy on different types of data-sets. In Figure 5 are reported two different benchmarks: "double_circles"(5a) and "swiss_roll" (5b), in which the two colors stand for the two different labels: {'0','1'}.



(a) (b)

Figure 5: Predictions of the REN-ODE (colors in the background) superimposed by the validation data called "double_circles" (a) and "swiss_roll" (b).

Continuous-Time System Identification REN-ODE has been tested for the system identification of a nonlinear pendulum in free evolution. The main goal of the experiment is to find a REN-ODE that is able to achieve a good approximation of the real physical system, i.e., given the same initial conditions, return the same outputs as the measurements of the pendulum's outputs. The pendulum is governed by the following equation:

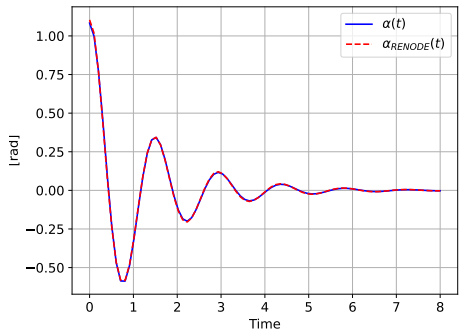
$$\ell\ddot{\alpha}(t) + \beta\dot{\alpha}(t) + g\sin\alpha(t) = 0 \tag{43}$$

where α is the angle of the pendulum with respect to the vertical axis, β is the viscous damping coefficient, g is the gravitational acceleration and ℓ is the length of the pendulum. The measurements $y(t)$ of the pendulum are the $\alpha(t)$ and $\dot{\alpha}(t)$, i.e., $y(t) = [\alpha(t) \;\; \dot{\alpha}(t)]^\top$. We want the output of the REN-ODE at each time instant to be as close as possible to the measurements of the pendulum. In order to do so, the loss function used for this task is the Mean Squared Error(MSE):

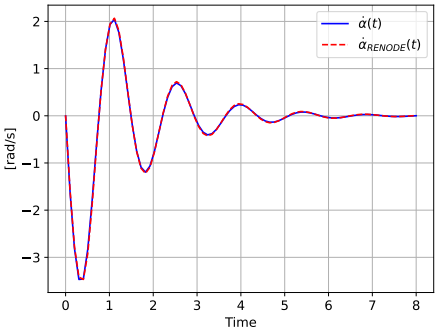
$$L(y, \hat{y}) = MSE(y, \hat{y}) = \frac{1}{\eta}\sum_{i=0}^{\eta}\sum_{t=0}^{T_{end}} \|y^i(t) - \hat{y}^i(t)\|^2 \tag{44}$$

where the loss is evaluated on a batch of η experiments and $y, \hat{y}$ are respectively the measured outputs vector and the predicted one. In order to use REN-ODE for system identification, the following strategy has been used: starting from N random initial conditions on the angle α and the velocity $\dot{\alpha}$, the N experiments of the mechanical system are simulated from T_0 up to time T_{end}; afterwards, η values are used as initial conditions of the neural network and then the model is let free to evolve up to the time T_{end}. Finally, the estimated outputs are compared with the previously simulated ones through the MSE loss function. In Figure 6 the trained model is tested comparing the evolution of the pendulum's states with the prediction

of the REN-ODE, starting from an initial state condition $(\alpha(0), \dot{\alpha}(0))$ not used during the training phase.



(a)



(b)

Figure 6: Validation of the REN-ODE, starting from an initial condition $(\alpha(0), \dot{\alpha}(0))$ not used during the training phase. In blue the actual trajectories and red the predicted one by the net.

Optimal Control The last considered use case of the REN-ODE is the optimal control of a dynamical system that consists of two robots that must reach two different targets, while avoiding any kind of collision and obstacles (4 "mountains"). Each robot has been modeled as a 2D point mass, subject to drag forces (e.g., friction) with state (p_t, q_t), where $p_t \in \mathbb{R}^2$ and $q_t \in \mathbb{R}^2$ represent the position and speed, respectively. For each agent:

$$\begin{bmatrix} \dot{p}_t \\ \dot{q}_t \end{bmatrix} = \begin{bmatrix} q_t \\ m^{-1}(-\mathcal{C}(q_t)q_t + F_t) \end{bmatrix} \tag{45}$$

where m is the mass of the robot, $F_t \in \mathbb{R}^2$ is the force control input and $\mathcal{C} : \mathbb{R}^2 \to \mathbb{R}$ is the *drag-force*, modeled as: $\mathcal{C}(q_t) = b_1 + b_2|q_t|$, with $b_1, b_2 \in \mathbb{R}^{1\times 2}$. The chosen control policy for this experiment is:

$$\sum_{k=0}^{T_{end}/T_s} l_{traj}(x_{t_k}, u_{t_k}) + l_{ca}(x_{t_k}) + l_{obst}(x_{t_k}) \tag{46}$$

where: l_{traj} penalizes the robots if they are not in the reference states and also minimizes the use of the inputs; l_{obst} "punishes" the robots if they move too close to the obstacles (the 4 "mountains") using the sum of 4 normal distribution functions with mean value in the peaks of the objects; l_{ca} penalizes the robots if their distance is smaller then a certain "safe value". The best result was obtained by a RREN-ODE with guarantees of strictly input passivity (with $\nu = 0.01$): in Figure 7 is reported one generated path for a random initial position of the robots. This model was able to avoid collisions for all the different combinations during the testing phase.

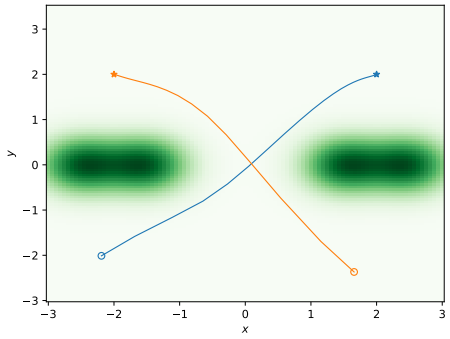


Figure 7: Path generated by the trained RREN-ODE strictly input passive with $\nu = 0.01$ starting from a random initial position of the robots.

8. Conclusions

Thanks to their flexibility, REN-ODEs can be used in different tasks in the control field. In this work, we have tested our new architecture, using it to identify the model of a nonlinear pendulum with different integration methods. Additionally, CREN-ODEs have been used in binary classification benchmarks from literature, obtaining great results. Finally, we implemented a RREN-ODE in a multi agent control scenario, where it was used as a regulator in order to be optimal with respect to a given policy function. RREN-ODE was able to achieve good performances, while trying to avoid obstacles and collisions between the agents. The guarantees of contractivity and robustness of this new class of neural networks, opens up the possibility in which REN-ODE's properties will be exploited, such as modeling of more challenging and complex systems like *reaction-diffusion systems* [1] or *continuous normalizing flows* [2]. Furthermore, future works will regard the use of REN-ODEs in distributed and decentralized cases, in which large-scale systems will be considered.

References

[1] M. Revay, R. Wang, and I. R. Manchester, "Recurrent Equilibrium Networks: Flexible Dynamic Models with Guaranteed Stability and Robustness," July 2021.

[2] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2018.

[3] M. Revay, R. Wang, and I. R. Manchester, "Lipschitz bounded equilibrium networks," 2020.

[4] S. Ruder, "An overview of gradient descent optimization algorithms," 2017.

[5] C. Verhoek, P. J. Koelewijn, R. Tóth, and S. Haesaert, "Convex incremental dissipativity analysis of nonlinear systems," *arXiv preprint arXiv:2006.14201*, 2020.

[6] A. Van der Schaft, *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

A. Proof of Theorem 4.1

Proof. We can define a function $V_\Delta(t)$ as:

$$V_\Delta(t) = \Delta x_t^\top P \Delta x_t. \tag{47}$$

Then, by left-multiplying and right-multiplying the inequality (21) with $\begin{bmatrix}\Delta x_t^\top & \Delta w_t^\top\end{bmatrix}$ and $\begin{bmatrix}\Delta x_t \\ \Delta w_t\end{bmatrix}$ respectively, it is possible to show that the following inequality is true:

$$\dot{V}_\Delta(t) < -\Gamma_t \leq 0. \tag{48}$$

Since $V_\Delta(t)$ is a quadratic form in the vector $\begin{bmatrix}\Delta x_t^\top & \Delta w_t^\top\end{bmatrix}^\top$ then it follows that, there exists a value $\alpha > 0$ such that $\dot{V}_\Delta(t) \leq -\alpha V_\Delta(t)$. This means that, according to Lyapunov exponential stability theorem, the incremental system is (globally) exponentially stable, and the system is *contractive*. □

B. Proof of Theorem 4.2

Proof. Considering the same function V_Δ as in (47), by left-multiplying and right-multiplying the inequality (23) with $\begin{bmatrix}\Delta x_t^\top & \Delta w_t^\top & \Delta u_t^\top\end{bmatrix}$ and $\begin{bmatrix}\Delta x_t^\top & \Delta w_t^\top & \Delta u_t^\top\end{bmatrix}^\top$ respectively, it is possible to show that the following inequality holds true:

$$\dot{V}_\Delta(t) - \begin{bmatrix}\Delta y_t^\top & \Delta u_t^\top\end{bmatrix}\begin{bmatrix}Q & S^\top \\ S & R\end{bmatrix}\begin{bmatrix}\Delta y_t \\ \Delta u_t\end{bmatrix} < -\Gamma_t \leq 0 \tag{49}$$

That is the *differentiated dissipation inequality* with respect to the storage function V_Δ, given the supply rate $s(\Delta u, \Delta y)$ as in (11). Thus, the system satisfies the incremental IQCs, as per Definition 2.5. □

C. How construct D_{22} in RREN-ODEs

We want to obtain a real D_{22} that guarantees that (30) holds true for any value of $\mathcal{R}$. We denote $s = \max(p, m)$; $X_3, Y_3 \in \mathbb{R}^{s\times s}$ and:

$$M = X_3^\top X_3 + Y_3 - Y_3^\top + \epsilon I, \tag{50}$$

$$Z = \left[(I - M)(I + M)^{-1}\right]_{p\times m}. \tag{51}$$

where $\epsilon > 0$ and $[V]_{p\times m}$ indicates the block matrix with the first p rows and m columns of a matrix V. Assuming $Q < 0$ and $R = R^\top$(always verified in the special choices reported in Section 2.5), Q and R can be factorized as:

$$L_Q^\top L_Q = -Q\,, \quad L_R^\top L_R = R - SQ^{-1}S^\top, \tag{52}$$

and finally:

$$D_{22} = -Q^{-1}S^\top + L_Q^{-1} Z L_R. \tag{53}$$

Constructing D_{22} as (53), it guarantees the inequality (30) always holds, for any choice of the IQC matrices, even for a null matrix S.