2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Neural Distributed Controllers with Port-Hamiltonian Structures

Muhammad Zakwan and Giancarlo Ferrari-Trecate

Abstract— **Controlling large-scale cyber-physical systems necessitates optimal distributed policies, relying solely on local real-time data and limited communication with neighboring agents. However, finding optimal controllers remains challenging, even in seemingly simple scenarios. Parameterizing these policies using Neural Networks (NNs) can deliver good performance, but their sensitivity to small input changes can destabilize the closed-loop system. This paper addresses this issue for a network of nonlinear dissipative systems. Specifically, we leverage well-established port-Hamiltonian structures to characterize deep distributed control policies with closed-loop stability guarantees and a finite $\mathcal{L}_2$ gain, regardless of specific NN parameters. This eliminates the need to constrain the parameters during optimization and enables training with standard methods like stochastic gradient descent. A numerical study on the consensus control of Kuramoto oscillators demonstrates the effectiveness of the proposed controllers.**

I. Introduction

Distributed control of *large-scale* systems presents formidable challenges even in seemingly basic scenarios due to the constrained flow of information in real-time. Particularly, Witsenhausen's counter-example [1] demonstrated that, even under apparently ideal conditions (i.e. linear dynamics, quadratic loss, and Gaussian noise), a nonlinear distributed control policy can outperform the best linear one. A stream of works such as [2] has provided necessary and sufficient condition, namely, Quadratic Invariance (QI), under which the distributed optimal controller is linear and corresponds to solving a convex optimization problem. However, real-world systems often violate QI assumptions due to inherent nonlinearities, non-convex control costs, or privacy limitations [3]. This necessitates venturing beyond linear control and exploring highly nonlinear distributed policies, such as those parametrized by Deep Neural Networks (DNNs).

DNNs have proved their capabilities in learning-enabled control [3]–[7], and system identification [8]–[15] of nonlinear dynamical systems. Indeed, NN control has been applied in diverse application domains, such as robotics [4], epidemic models [16], safe path planning [6], and Kuramoto oscillators [17]. Existing approaches to NN control design also include modelling the system under control as a NN from data [18]–[21]. Nevertheless, NNs can be susceptible to small changes in their inputs [22]. This fragility can easily translate to neural control policies, potentially jeopardizing the stability of closed-loop systems [10]. Moreover, the large number of parameters and intricate interconnections within NNs make it difficult to verify them for safety certificates and use them in large-scale safety-critical applications [6].

In this paper, we leverage well-established port-Hamiltonian (pH) system framework [23] to parametrize distributed DNN control policies that are inherently endowed with a finite $\mathcal{L}_2$ gain regardless of the choice of trainable parameters. This results in an unconstrained optimization problem for DNN control design solvable using standard gradient-based methods such as stochastic gradient descent or its variants. This eliminates the need for computationally expensive approaches such as projection of weight matrices or constrained optimization techniques. Therefore, if the underlying system to be controlled is dissipative, the proposed DNN controllers guarantee closed-loop $\mathcal{L}_2$ stability both during and after the training. Moreover, the learned distributed policies are optimal in the sense that they minimize an arbitrary nonlinear cost function over a finite horizon.

Related work DNNs have shown promise in designing both static and dynamic distributed control policies for large-scale systems. Notably, Graph Neural Networks (GNNs) have achieved impressive performance in applications like vehicle flocking and formation flying [24]–[27] thanks to their inherent scalable structure. However, guaranteeing stability with general GNNs remains challenging, often requiring restrictive assumptions like linear, open-loop stable system dynamics or sufficiently small Lipschitz constants [27]. Such limitations can be impractical, potentially leading to system failures during the training phase before an optimal policy can be found [28], [29]. Some remedies to rectify this problem include improving an initial known safe policy iteratively, while imposing the constraint that the initial *region of attraction* does not shrink [30]–[32], and leveraging integral quadratic constraints to enforce closed-loop stability of DNN controllers [33]. However, these approaches explicitly constrain the DNN weights, which may lead to infeasibility or hinder the closed-loop performance. In contrast, our proposed method based on free parameterizations provides the same scalability as GNNs without imposing any constraints on the weight matrices to guarantee closed-loop stability. Although previous work explored stable-by-design control based on mechanical energy conservation [34], [35], these methods are limited to specific systems (e.g., SE(3) dynamics). On the other hand, our approach applies to a wider range of nonlinear systems.

Recently, the notion of *free parametrization* has emerged for learning-enabled control, where an NN controller is trained to ensure its weight matrices satisfy specific constraints (e.g., semi-definite programs) *by design*. This allows us to bypass computationally expensive post-verification rou-

This research is supported by the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40 180545).

Authors are with the Institute of Mechanical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, email: {muhammad.zakwan, giancarlo.ferraritrecate}@epfl.ch

979-8-3503-1632-2/24/$31.00 ©2024 IEEE

tines. Based on this approach, the framework of Recurrent Equilibrium Networks (RENs) has been proposed in [10]. RENs are a class of neural discrete-time nonlinear dynamical models that ensure built-in stability and robustness. Notably, they possess the unique property of satisfying desired integral quadratic constraints regardless of their weight matrices. Despite their flexibility, RENs face several limitations. Firstly, they are restricted to capturing dynamics with quadratic storage functions, limiting their expressiveness for complex systems. Secondly, the free parameterization approach in [10], [36] cannot be directly applied to distributed systems where sparsity patterns in weight matrices are crucial. In contrast, our NN framework based on pH structures offers several advantages. It allows the use of arbitrary nonlinear storage functions to capture more complex dynamics. Additionally, it seamlessly integrate desired sparsity patterns into the weight matrices, enhancing flexibility without compromising stability and performance. Building on RENs, the work [37] presents an unconstrained parametrization approach for interconnecting subsystems with finite $\mathcal{L}_2$ gain, while guaranteeing the $\mathcal{L}_2$ stability of the overall system. However, this approach is limited to quadratic storage functions for subsystems, constraining the flexibility and generalization. Although [3] presented a similar distributed NN framework based on pH systems that ensure passivity by design but not a finite $\mathcal{L}_2$ gain for the closed-loop system which is instead our main result. Unlike passivity, a finite $\mathcal{L}_2$ gain guarantees stability even in the presence of external disturbances or modeling errors, which is crucial for safe operation in uncertain environments [23], [38].

Contributions The main contributions of this paper can be summarized as follows:

1) We provide a free parametrization of distributed controllers that can seamlessly incorporate sparsity in their weight matrices and are inherently endowed with a finite $\mathcal{L}_2$ gain.
2) Our approach overcomes the limitation of being restricted to specific storage functions (e.g., quadratic), enabling its application to a broader range of nonlinear control problems.
3) We demonstrate the efficacy of our learning-enabled controllers on a benchmark consensus problem for Kuramoto oscillators.

Organization: Following the Introduction, Section II provides some preliminaries and the problem formulation. In Section III, we provide a free parametrization of neural distributed controllers via Hamiltonian structures endowed with a finite $\mathcal{L}_2$ gain regardless of the choice of weight matrices. Finally, the performance evaluation of our NN controllers is conducted in Section IV, whereas Section V concludes the paper.

Notation: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with nodes $\mathcal{V} = \{1, \dots, N\}$ and edges $\mathcal{E}$, and let $\mathcal{P} \in \{0,1\}^{N\times N}$ be the corresponding adjacency matrix. For a binary mask $\mathcal{M} \in \{0,1\}^{m\times n}$, we denote $\boldsymbol{W} \in \texttt{blkSparse}(\mathcal{M})$ if $\boldsymbol{W}$ is a block matrix and $\mathcal{M}_{i,j} = 0 \Rightarrow \boldsymbol{W}_{i,j} = 0$. $\boldsymbol{A} = \texttt{blkdiag}(A_i)$ represents a block-diagonal matrix with matrices $A_0, A_1, \dots, A_i$ on the diagonal. The set of non-negative real numbers is $\mathbb{R}_+$ and the standard Euclidean 2-norm is denoted by $\|\cdot\|$. We represent the set of $\mathbb{R}^n$-valued Lebesgue square-integrable functions by $\mathcal{L}_2^n := \{v : [0,\infty) \to \mathbb{R}^n | \|v\|_2^2 := \int_0^\infty v(t)^\top v(t)dt < \infty\}$. We omit the dimension n whenever it is clear from the context. Then, for any two $v, w \in \mathcal{L}_2^n$, we denote the $\mathcal{L}_2^n$-inner product as $\langle v, w\rangle := \int_0^\infty v(t)^\top w(t)dt$. Define the truncation operator $(P_\mathcal{T} v)(t) := v(t)$ for $t \le \mathcal{T}$; $(P_\mathcal{T} v)(t) := 0$ for $t > \mathcal{T}$, and the extended function space $\mathcal{L}_{2e}^n := \{v : [0,\infty) \to \mathbb{R}^n | P_\mathcal{T} v \in \mathcal{L}_2, \forall \mathcal{T} \in [0,\infty)\}$. For any linear space $\mathcal{U}$ endowed with a norm $\|\cdot\|_\mathcal{U}$, we define a Banach space $\mathcal{L}_{2e}(\mathcal{U})$ that consists of all measurable functions $f : \mathbb{R}_+ \mapsto \mathcal{U}$ such that $\int_0^\infty \|f(t)\|_\mathcal{U}^2 dt < \infty$. Throughout this paper, a system will be specified by an input–output map $\Sigma : \mathcal{L}_{2e}^m \to \mathcal{L}_{2e}^p$ satisfying $\Sigma(0) = 0$. Given two systems Σ_1 and Σ_2, the standard negative feedback configuration between them is denoted by $\Sigma_1 \|_f \Sigma_2$, see Fig. 2. The maximal eigenvalue of a matrix A is represented by $\bar{\lambda}(A)$.

II. Preliminaries and Problem Formulation

We consider a network Σ_s of $N \in \mathbb{N}$ coupled nonlinear subsystems, each endowed with a feedback control policy. Let $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$ represents the graph associated with the couplings among subsystems, and let $\mathcal{P}_s$ be its corresponding adjacency matrix. Then, each subsystem is governed by

$$\Sigma_{s,i} : \begin{cases} \dot{x}_i(t) = f_i\left(x_i(t), \breve{x}_i(t), u_i(t)\right), & \text{(1a)} \\ y_i(t) = h_i(x_i(t)), \qquad \forall i \in \mathcal{V}, & \text{(1b)} \end{cases}$$

where $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is the state, $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ is the input, and $y_i \in \mathcal{Y}_i \subseteq \mathbb{R}^{p_i}$ is the output of the subsystem $\Sigma_{s,i}$, respectively. We define $\breve{x}_i$ as a stacked vector of states of the 1-hop neighbors of subsystem i according to $\mathcal{G}_s$, i.e. all subsystems that influence x_i. We assume there exists a unique solution trajectory $x_i(\cdot)$ on the infinite time interval $[0,\infty)$ of the differential equations (1a) for all initial conditions $x_i(0) \in \mathcal{X}_i$ and $u_i(\cdot) \in \mathcal{L}_{2e}(\mathcal{U}_i)$, and $y_i(\cdot) \in \mathcal{L}_{2e}(\mathcal{Y}_i)$. We assume that the distributed system Σ_s is dissipative according to the following definition.

Definition 1 (Dissipativity, [23]): The subsystem $\Sigma_{s,i}$ is called dissipative w.r.t. to a supply rate $s_i : \mathcal{U}_i \times \mathcal{Y}_i \mapsto \mathbb{R}$, if there exists a smooth storage function $V_i : \mathcal{X}_i \mapsto \mathbb{R}_+$ such that

$$\dot{V}_i(x_i(t)) \le s_i(u_i(t), y_i(t)), \quad \forall t \in \mathbb{R}_+ ,$$

or equivalently,

$$V_i(x_i(\tau)) - V_i(x_i(0)) \le \int_0^\tau s_i(u_i(t), y_i(t))dt ,$$

for every input signal $u_i(t) \in \mathcal{U}_i$, output signal $y_i(t) \in \mathcal{Y}_i$ and every $\tau \ge 0$. Moreover, the choice of supply rate leads to different notions of dissipativity, for instance,

- if $p_i = m_i$, and $s_i(u_i(t), y_i(t)) = u_i(t)^\top y_i(t)$, then system $\Sigma_{s,i}$ is passive;

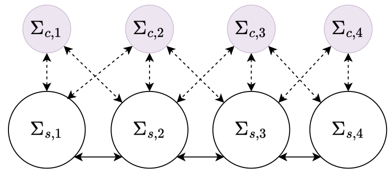


Fig. 1: An example of a large-scale system Σ_s and a distributed controller Σ_c for $N = 4$. The solid lines represent interactions between the subsystems of Σ_s, and the dashed lines represent the flow of information between the system Σ_s and the controller Σ_c.

- if $p_i = m_i$, and $s_i(u_i(t), y_i(t)) = u_i(t)^\top y_i(t) + \epsilon\|y_i(t)\|$, then system $\Sigma_{s,i}$ is ϵ-output strictly passive for $\epsilon > 0$;
- if $s_i(u_i(t), y_i(t)) = \gamma^2\|u_i(t)\| + \|y_i(t)\|$, then system $\Sigma_{s,i}$ has finite $\mathcal{L}_2$ gain, i.e. $\|y_i(t)\| \leq \gamma\|u_i(t)\| + b$ for some non-negative constants γ, b.

Recall that ϵ-output strict passivity also implies a finite $\mathcal{L}_2$ gain not larger than $1/\epsilon$ [23]. Note that the storage function $V(\cdot)$ can be interpreted as the stored "energy" in the system w.r.t. a single point of neutral storage (minimum energy).

The distributed control of large-scale systems presents a major challenge: local controllers at each subsystem $u_i(t)$ can only access real-time information from a limited set of neighbors, dictated by a communication network $\mathcal{G}_c$. This network is represented by an adjacency matrix $\mathcal{P}_c \in \{0,1\}^{N\times N}$ where $\mathcal{P}_{c_{i,i}} = 1$ for every $i \in \mathcal{V}$. In this paper, our goal is to develop distributed dynamic feedback controller Σ_c represented by the pairs $(\chi_i(\cdot), \pi_i(\cdot)), i \in \mathcal{V}$ defining local controllers

$$\Sigma_{c,i} : \begin{cases} \dot{\xi}_i(t) = \chi_i(\xi_i(t), \breve{y}_i(t)), & \xi_i \in \Xi_i \subseteq \mathbb{R}^{q_i}, \\ u_i(t) = \pi_i(\xi_i(t), \breve{y}_i(t)) , \end{cases} \tag{2}$$

where ξ_i is the state of $\Sigma_{c,i}$, and $\breve{y}_i(t)$ is a stacked vector of outputs from the neighbouring subsystems based on the communication graph $\mathcal{G}_c$. An example of a communication graph between the controller Σ_c and the distributed systems Σ_s is illustrated in Fig. 1. Moreover, the control policies parametrized by Σ_c should be *optimal* in the sense that they minimize an arbitrary real-valued cost function

$$c(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \frac{1}{T}\int_0^T \ell(\boldsymbol{x}(t), \boldsymbol{u}(t))dt \tag{3}$$

for a finite horizon $T \in \mathbb{R}_+$, where c is differentiable almost everywhere. The bold-faced signals $\boldsymbol{x}(t) = [x_1^\top, \ldots, x_N^\top]^\top, \boldsymbol{u}(t) = [u_1^\top, \ldots, u_N^\top]^\top$ represent concatenated local states and local inputs, respectively. Finally, we assume that the set of tuples $\{(\chi_i(\cdot), \pi_i(\cdot))\}, i \in \mathcal{V}$ is parametrized by some NNs with trainable parameters $\theta_i \in \mathbb{R}^{d_i}$ for $i \in \mathcal{V}$. We define $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$.

Besides designing optimal control policies, ensuring the stability of the closed-loop $\Sigma_s\|_f\Sigma_c$ formed by the distributed system Σ_s and the NN controller Σ_c is equally crucial. To address this issue, we focus on achieving $\mathcal{L}_2$ stability for the closed-loop system $\Sigma_s\|_f\Sigma_c$, leveraging the following results.

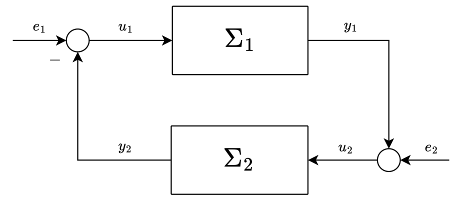


Fig. 2: Standard feedback interconnection $\Sigma_1\|_f\Sigma_2$.

Theorem 1 ([23]): Consider the closed-loop system $\Sigma_1\|_f\Sigma_2$ given in Fig. 2.

- (small gain condition) Assume the existence of the $\mathcal{L}_2$ gains $\mathcal{L}_2(\Sigma_1) \leq \gamma_1$, and $\mathcal{L}_2(\Sigma_2) \leq \gamma_2$. Then, the closed-loop system $\Sigma_1\|_f\Sigma_2$ is stable with an $\mathcal{L}_2$ gain $\gamma_1.\gamma_2 < 1$;
- (strict output passivity) Assume that, for any $e_1 \in \mathcal{L}_{2e}(\mathcal{U}_1)$ and $e_2 = 0$, $\Sigma_1 : \mathcal{L}_{2e}(\mathcal{U}_1) \to \mathcal{L}_{2e}(\mathcal{Y}_1)$ is ϵ_1-output strictly passive, and $\Sigma_2 : \mathcal{L}_{2e}(\mathcal{U}_2) \to \mathcal{L}_{2e}(\mathcal{Y}_2)$ is passive. Then, $\Sigma_1\|_f\Sigma_2$ for $e_2 = 0$ with input e_1 and output y_1 has $\mathcal{L}_2$-gain $\leq 1/\epsilon_1$. ■

Our goal is to train control policies for large-scale systems that address three key requirements:

i) Limited information access: The policies must operate with restricted local information.
ii) Optimal performance: They achieve optimal behavior by empirically minimizing an arbitrary user-defined cost function.
iii) Guaranteed stability: The closed-loop system is $\mathcal{L}_2$ stable.

This can be formulated as the following optimization program

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{S}\sum_{k=1}^{S} c(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\theta}, \boldsymbol{x}_0^k) \tag{4}$$

$$\text{s.t.} \quad \text{system dynamics } \Sigma_s$$

$$\dot{\xi}_i(t) = \chi_i(\xi_i, \breve{y}_i(t), \theta_i), \tag{5}$$

$$u_i(t) = \pi_i(\xi_i, \breve{y}_i(t), \theta_i), \quad \forall i \in \mathcal{V}, \tag{6}$$

$$(5) - (6) \text{ has a finite } \mathcal{L}_2 \text{ gain}, \forall \boldsymbol{\theta} \in \mathbb{R}^{\boldsymbol{d}} , \tag{7}$$

where S is the number of given initial conditions for Σ_s. The primary challenge of this optimization problem lies in finding the parameters $\boldsymbol{\theta}$ such that the distributed controller (5)-(6) has a finite $\mathcal{L}_2$ gain without jeopardizing the standard NN training routines and without increasing the computational complexity. The next Section presents a novel method that tackles this challenge effectively.

Remark 1 (Passivity by design): While achieving passivity by design for the closed-loop system is considered in [3], it may not always ensure stability, especially when controlled system interacts with a passive, but else completely unknown environment. In fact, the converse of the passivity theorem tells us that the controlled system must be output strictly passive as seen from the interaction port of the controlled system with the environment [38].

III. Neural $\mathcal{L}_2$-stable Hamiltonian Controllers

To address the $\mathcal{L}_2$ constraint (7) in the optimization program (4)-(7), two common approaches [6] involve constrained optimization or projection of $\boldsymbol{\theta}$ onto a set $\Theta_{\mathcal{L}_2}$ such that the NN controller Σ_c has a finite $\mathcal{L}_2$ gain. However, these methods can be computationally burdensome, hence limiting the class of controllers that can be used. To circumvent this, we propose a *free parametrization* approach which involves designing a class of input-output operators that inherently possess a finite $\mathcal{L}_2$ gain for any choice of weight matrices $\boldsymbol{\theta}$. This enables us to seamlessly employ unconstrained optimization methods such as stochastic gradient descent and its variants to solve (4)-(7). To achieve this, we leverage the well-established port-Hamiltonian framework [23] to parametrize controllers Σ_c with guaranteed finite $\mathcal{L}_2$ gains.

Consider the following neural distributed pH controller with N sub-controllers endowed with some trainable parameters

$$\Sigma_{pH}: \begin{cases} \dot{\boldsymbol{\xi}}(t) = [\boldsymbol{J}_c - (\alpha \boldsymbol{I} + \boldsymbol{\Lambda})] \dfrac{\partial H_c}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}, \theta) + \boldsymbol{G}_c \boldsymbol{y}(t) \\ \boldsymbol{u}(t) = \boldsymbol{G}_c^\top \dfrac{\partial H_c}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}, \theta) \ , \end{cases} \tag{8}$$

where $\boldsymbol{\xi} \in \Xi \subseteq \mathbb{R}^{n_c}, \boldsymbol{u} \in \mathcal{U}_c \subseteq \mathbb{R}^m, \boldsymbol{y} \in \mathcal{Y}_c \subseteq \mathbb{R}^m$ are stacked vectors of NN controllers' states, outputs, and inputs, respectively. The interconnection matrix $\boldsymbol{J}_c = -\boldsymbol{J}_c^\top = \texttt{blkdiag}(J_i)$ is skew-symmetric. The dissipation rate of Σ_{pH} is determined by the damping matrix $\alpha \boldsymbol{I} + \boldsymbol{\Lambda}$, where $\alpha \in \mathbb{R}_+$, and $\boldsymbol{\Lambda} = \text{diag}(e^{\boldsymbol{d}}) \in \mathbb{R}_+^n$ is a diagonal matrix for some free vector $\boldsymbol{d} \in \mathbb{R}^{n_c}$. The input matrix $\boldsymbol{G}_c = \texttt{blkSparse}(\mathcal{P}_c)$ is full rank, where $\mathcal{P}_c$ is the underlying adjacency matrix of the communication topology.[1] Moreover, the "energy-like" Hamiltonian function $H_c : \mathbb{R}^{n_c} \to \mathbb{R}$ of Σ_{pH} is the algebraic sum of all N controllers' energies H_i, i.e. $H_c(\boldsymbol{\xi}) = \sum_{i \in \mathcal{V}} H_i(\xi_i(t))$, where we assume that all functions H_i are continuously differentiable and radially unbounded.

Our main result concerns a free parametrization of the NN controller Σ_{pH} that guarantees a finite $\mathcal{L}_2$ gain regardless of the choice of its trainable parameters. These parameters, collectively denoted by $\boldsymbol{\theta}$, encompass the weight matrices $\{\boldsymbol{J}_c, \boldsymbol{\Lambda}, \boldsymbol{G}_c, \theta\}$.

Theorem 2: Given a constant $\epsilon > 0$, let $\alpha = \epsilon \bar{\lambda}(\boldsymbol{G}_c \boldsymbol{G}_c^\top)$. Then, the NN controller Σ_{pH}

- is ϵ-output strictly passive, and
- has a finite $\mathcal{L}_2$-gain $\leq 1/\epsilon$. ■

The detailed proof of Theorem 2 is provided in the accompanying technical report [39]. In simple words, Theorem 2 implies that for any free choice of trainable parameters $\boldsymbol{\theta}$, one can always choose sufficiently large damping α such that the controller Σ_c is ϵ-output strictly passive, and consequently, the map from $\boldsymbol{y}(t) \mapsto \boldsymbol{u}(t)$ has a finite $\mathcal{L}_2$ gain. Therefore, one can leverage Theorem 2 and invoke Theorem 1 to ensure closed-loop stability in cases where the system Σ_s is passive (strict output passivity) or has a finite $\mathcal{L}_2$ gain (small gain condition).

While we assume that the large-scale system Σ_s is passive from $\boldsymbol{u}(t)$ to $\boldsymbol{y}(t)$ in our simplified setting, passivity truly holds only if all subsystems are individually passive (from $u_i(t)$ to $y_i(t)$) and the interconnection is power-conserving (e.g. skew-symmetric [40]). For deeper insights into preserving passivity and $\mathcal{L}_2$ gain in large-scale interconnected systems, refer to [40]. Moreover, we defer the reader to [39] for a note on the training methodologies of these controllers.

Remark 2 (Selection of Hamiltonian): We impose minimal restrictions on the Hamiltonian function $H_c(\boldsymbol{\xi}, \theta)$, i.e., differentiability and radial unboundedness. This flexibility allows for diverse choices, including simple quadratic functions, multi-layer perceptrons (MLPs), or even Hamiltonian deep neural networks (as in [41], [42] for representing H_c). Importantly, our results hold irrespective of the choice.

Remark 3 (Communication among sub-controllers): Note that our framework can seamlessly incorporate communication graphs among the sub-controllers without loss of generality. In fact, the work [3] provides a systematic approach to interconnect sub-controllers while preserving the dissipativity of the closed-loop. For specific details and an example, we defer the readers to [3, Theorem 3].

IV. Synchronization in Kuramoto Oscillators

To showcase the efficacy of our NN control framework, we consider the problem of synchronization in Kuramoto oscillator model. Indeed, this problem is pervasive for investigating collective synchronous behaviors in several applications. For example, locking of circuit oscillators, frequency synchronization in power grids [43], collective motion of self-propelled vehicles, and opinion synchronization in social networks. The interested readers are deferred to [44] and references therein for more details.

A Kuramoto model consists of a population of N oscillators whose dynamics are

$$\dot{\vartheta}_i = \omega_i + \frac{K u_i(t)}{N} \sum_{j=1}^{N} \mathcal{P}_{ij} \sin(\vartheta_j - \vartheta_i), \ i \in \mathcal{V} \ , \tag{9}$$

where ϑ_i is the phase and ω_i is the natural frequency of $i-$th oscillator, respectively. Moreover, K is the coupling strength and $\mathcal{P}_{ij}$ are the adjacency matrix components of the underlying (undirected) network. While the NN control of Kuramoto oscillators has been considered in [17], their approach lacks closed-loop stability guarantees, which might lead to undesirable system behavior.

Let all the oscillators at $t = 0$ be initialized in the set $\mathcal{D} := \{\vartheta_i, \vartheta_j \text{ s.t. } |\vartheta_i - \vartheta_j| < \frac{\pi}{2} \ \forall i, j = \mathcal{V}\}$. Then, by constructing the dynamics of the angular frequencies and differentiating the Kuramoto model (9), one obtains

$$\ddot{\vartheta}_i = \frac{K u_i(t)}{N} \sum_{j=1}^{N} \mathcal{P}_{ij} \cos(\vartheta_j - \vartheta_i)(\dot{\vartheta}_j - \dot{\vartheta}_i) \ . \tag{10}$$

[1] Decentralized control is achieved by setting $\boldsymbol{G}_c = \texttt{blkdiag}(G_i)$ in (8), making each sub-controller independent of the state if other subsystems or controllers.

By the change of variables $\dot{\vartheta}_i = x_i$ in (10), we have

$$\dot{x}_i = \nu,\ \nu = \frac{Ku_i(t)}{N}\sum_{j=1}^{N}\mathcal{P}_{ij}g_{ji}(x_j - x_i)\ , \qquad (11)$$
$$y_i = x_i\ ,$$

where $g_{ji} = \cos(\vartheta_j - \vartheta_i)$. One can show that the system (11) is passive w.r.t. the storage function $V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\boldsymbol{x}$, where $\boldsymbol{x} = [x_1,\ \dots x_N]^\top$ is the concatenated vector representing the angular rates of the oscillators [45]. Our goal is the phase-synchronization of (11) at some final time $T > 0$, that is,

$$|x_i(T) - x_j(T)| = 0 \text{ for } \mathcal{P}_{i,j} = 1\ ,$$

while guaranteeing the closed-loop stability both during and after the training. Thus, our objective can be formulated as the following optimization problem

$$\min_{\boldsymbol{\theta}} c(\boldsymbol{x}(t),\boldsymbol{u}(t);\boldsymbol{\theta},\boldsymbol{x}_0) \text{ s.t. the closed-loop is } \mathcal{L}_2 \text{ stable}$$
$$c(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{x}_0) = \frac{1}{2}\int_0^T \sum_{i,j}\mathcal{P}_{ij}\sin^2(\vartheta_j(s) - \vartheta_i(s)) + \beta\|\boldsymbol{u}(s)\|_2^2 ds\ ,$$

where the regularization term β.

We choose the parametrization of the neural pH controller (8) with trainable parameters of appropriate dimensions, $\boldsymbol{J}_c = \texttt{blkdiag}(J_i)$, $\boldsymbol{G}_c = \texttt{blkSparse}(\mathcal{P})$, and the Hamiltonian function H_c as

$$H_c(\boldsymbol{\xi}) = \log\left(\cosh[\texttt{blkdiag}(K_i)\boldsymbol{\xi}]\right)^\top \mathbb{1}\ , \qquad (12)$$

where $\mathbb{1}$ is a vector of all ones. Note that unlike RENs, where the storage function is always quadratic, our choice of Hamiltonian function is nonlinear. Moreover, one can also analytically compute the closed-form solution of the Jacobian of (12) for each sub-controller as

$$\frac{\partial H_i}{\partial \xi_i}(\xi,\theta_i) = K_i^\top \tanh(K_i\xi_i),\ \forall i \in \mathcal{V}\ .$$

We trained the NN controller with the standard adjoint method and Forward Euler as the discretization scheme [46]. The training is performed with 500 epochs using Adam [47] with a learning rate of $5e-3$. We choose $\epsilon = 0.85$ and the finite horizon $T = 3.0$. To measure the degree of synchronization in the network, we introduce the metric

$$r(t) := N^{-1}\sqrt{\sum_{i,j}\cos(\vartheta_j(t) - \vartheta_i(t))},\ \forall i,j \in \mathcal{V}\ .$$

Note that a value of $r(t) = 1$ indicates that all oscillators have the same phase. The closed-loop response of (11) and (8) after the training is provided in Fig. 3, which demonstrates the synchronization for different communication topologies. Particularly, we study the controller performance on a complete graph (grey lines, Fig. 3a), an Erdös–Rényi network $\mathcal{G}(N,p)$ with $p = 0.3$ (blue lines, Fig. 3b), a square lattice (green lines, Fig. 3c), and a Watts–Strogatz network with degree $k = 5$ and a rewiring probability of 0.3 (red lines, Fig. 3d) taken from [16]. All networks consist of $N = 64$ oscillators. As shown, the NN controller effectively drives the oscillators to consensus (all $r(t)$ reach 1). Furthermore, we observe that consensus is maintained even after the finite-horizon $T = 3$ used for optimization, demonstrating the closed-loop stability.[2]

V. Conclusion

Neural distributed control of large-scale nonlinear systems can pose several challenges, such as guaranteeing closed-loop stability in an uncertain environment. To tackle this issue, we have proposed a free parametrization of neural distributed controllers via Hamiltonian structures that preserve closed-loop stability and guarantee a finite $\mathcal{L}_2$ gain, regardless of NN parameters, for arbitrarily large networks of nonlinear dissipative systems. We demonstrated that near-optimal performance can be achieved by parametrizing deep nonlinear storage functions for the controllers. Moreover, these NN structures can be leveraged for nonlinear system identification from data, where the identified neural models are stable by design and have a finite $\mathcal{L}_2$ gain. Further efforts will be devoted to exploring discretization schemes to preserve the $\mathcal{L}_2$ gain and, consequently, implementing the NN controllers on digital systems.

References

[1] H. S. Witsenhausen, "A counterexample in stochastic optimum control," *SIAM Journal on Control*, vol. 6, no. 1, pp. 131–147, 1968.

[2] L. Lessard and S. Lall, "Quadratic invariance is necessary and sufficient for convexity," in *IEEE American Control Conference (ACC)*, 2011, pp. 5360–5362.

[3] L. Furieri, C. L. Galimberti, M. Zakwan, and G. Ferrari-Trecate, "Distributed neural network control with dependability guarantees: a compositional port-hamiltonian approach," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 571–583.

[4] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[5] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021.

[6] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods," *arXiv preprint arXiv:2202.11762*, 2022.

[7] L. Furieri, C. L. Galimberti, and G. Ferrari-Trecate, "Learning to boost the performance of stable nonlinear systems," *IEEE Open Journal of Control Systems*, pp. 1–16, 2024.

[8] T. X. Nghiem, J. Drgoňa, C. Jones, Z. Nagy, R. Schwan, B. Dey, A. Chakrabarty, S. Di Cairano, J. A. Paulson, A. Carron *et al.*, "Physics-informed machine learning for modeling and control of dynamical systems," *arXiv preprint arXiv:2306.13867*, 2023.

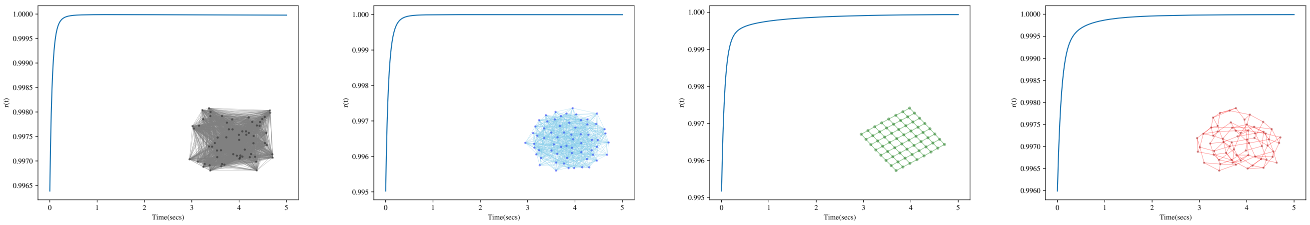
[9] C. Verhoek, G. I. Beintema, S. Haesaert, M. Schoukens, and R. Tóth, "Deep-learning-based identification of LPV models for nonlinear systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 3274–3280.

[10] 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.

[11] R. Wang, N. H. Barbara, M. Revay, and I. R. Manchester, "Learning over all stabilizing nonlinear controllers for a partially-observed linear system," *IEEE Control Systems Letters*, vol. 7, pp. 91–96, 2022.

[12] M. Scandella, M. Bin, and T. Parisini, "Kernel-based identification of incrementally input-to-state stable nonlinear systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5127–5132, 2023.

[2]Our code is available at https://github.com/DecodEPFL/Neural-Distributed-Controllers.git



(a) A complete graph. (b) An Erdos-Renyi graph. (c) A Square Lattice. (d) A Watt-Strogatz.

Fig. 3: The consensus metric $r(t)$ for the closed-loop with different communication topologies exhibiting consensus.

[13] M. Zakwan, L. Di Natale, B. Svetozarevic, P. Heer, C. N. Jones, and G. F. Trecate, "Physically consistent neural odes for learning multi-physics systems," *arXiv preprint arXiv:2211.06130*, 2022.

[14] L. Di Natale, M. Zakwan, P. Heer, G. F. Trecate, and C. N. Jones, "Simba: System identification methods leveraging backpropagation," *arXiv preprint arXiv:2311.13889*, 2023.

[15] L. Di Natale, M. Zakwan, B. Svetozarevic, P. Heer, G. F. Trecate, and C. N. Jones, "Stable linear subspace identification: A machine learning approach," *arXiv preprint arXiv:2311.03197*, 2023.

[16] T. Asikis, L. Böttcher, and N. Antulov-Fantulin, "Neural ordinary differential equation control of dynamics on graphs," *Physical Review Research*, vol. 4, no. 1, p. 013221, 2022.

[17] L. Böttcher, N. Antulov-Fantulin, and T. Asikis, "AI pontryagin or how artificial neural networks learn to control dynamical systems," *Nature communications*, vol. 13, no. 1, p. 333, 2022.

[18] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019.

[19] L. B. Armenio, E. Terzi, M. Farina, and R. Scattolini, "Model predictive control design for dynamical systems learned by echo state networks," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 1044–1049, 2019.

[20] F. Bonassi, M. Farina, J. Xie, and R. Scattolini, "On recurrent neural networks for learning-based control: recent results and ideas for future developments," *Journal of Process Control*, vol. 114, pp. 92–104, 2022.

[21] E. Terzi, F. Bonassi, M. Farina, and R. Scattolini, "Learning model predictive control with long short-term memory networks," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8877–8896, 2021.

[22] M. Zakwan, L. Xu, and G. Ferrari-Trecate, "Robust classification using contractive hamiltonian neural odes," *IEEE Control Systems Letters*, vol. 7, pp. 145–150, 2022.

[23] A. van der Schaft, *L2-Gain and Passivity Techniques in Nonlinear Control*, 3rd ed., ser. Communications and Control Engineering. Cham: Springer International Publishing, 2017.

[24] F. Yang and N. Matni, "Communication topology co-design in graph recurrent neural network based distributed control," *arXiv preprint arXiv:2104.13868*, 2021.

[25] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro, "Learning decentralized controllers for robot swarms with graph neural networks," in *Conference on robot learning*. PMLR, 2020, pp. 671–682.

[26] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, "Graph policy gradients for large scale robot control," in *Conference on robot learning*. PMLR, 2020, pp. 823–834.

[27] F. Gama and S. Sojoudi, "Graph neural networks for distributed linear-quadratic control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 111–124.

[28] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *arXiv preprint arXiv:2108.06266*, 2021.

[29] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[30] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," *Advances in Neural Information Processing Systems 30*, vol. 2, pp. 909–919, 2018.

[31] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *Conference on Robot Learning*. PMLR, 2018, pp. 466–476.

[32] T. Koller, F. Berkenkamp, M. Turchetta, and A. Krause, "Learning-based model predictive control for safe exploration," in *2018 IEEE conference on decision and control (CDC)*. IEEE, 2018, pp. 6059–6066.

[33] P. Pauli, J. Köhler, J. Berberich, A. Koch, and F. Allgöwer, "Offset-free setpoint tracking using neural network controllers," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 992–1003.

[34] S. Abdulkhader, H. Yin, P. Falco, and D. Kragic, "Learning deep energy shaping policies for stability-guaranteed manipulation," *IEEE Robotics and Automation Letters*, 2021.

[35] T. Duong and N. Atanasov, "Hamiltonian-based neural ODE networks on the SE (3) manifold for dynamics learning and control," *arXiv preprint arXiv:2106.12782*, 2021.

[36] D. Martinelli, C. L. Galimberti, I. R. Manchester, L. Furieri, and G. Ferrari-Trecate, "Unconstrained parametrization of dissipative and contracting neural ordinary differential equations," *arXiv preprint arXiv:2304.02976*, 2023.

[37] L. Massai, D. Saccani, L. Furieri, and G. Ferrari-Trecate, "Unconstrained learning of networked nonlinear systems via free parametrization of stable interconnected operators," *arXiv preprint arXiv:2311.13967*, 2023.

[38] S. Z. Khong and A. van der Schaft, "On the converse of the passivity and small-gain theorems for input–output maps," *Automatica*, vol. 97, pp. 58–63, 2018.

[39] M. Zakwan and G. Ferrari-Trecate, "Neural distributed controllers with port-hamiltonian structures," *arXiv preprint arXiv:2403.17785*, 2024.

[40] M. Arcak, C. Meissen, and A. Packard, *Networks of dissipative systems: compositional certification of stability, performance, and safety*. Springer, 2016.

[41] C. L. Galimberti, L. Furieri, L. Xu, and G. Ferrari-Trecate, "Hamiltonian deep neural networks guaranteeing nonvanishing gradients by design," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 3155–3162, 2023.

[42] M. Zakwan, M. d'Angelo, and G. Ferrari-Trecate, "Universal approximation property of hamiltonian deep neural networks," *IEEE Control Systems Letters*, 2023.

[43] F. Dörfler and F. Bullo, "Synchronization in complex networks of phase oscillators: A survey," *Automatica*, vol. 50, no. 6, pp. 1539–1564, 2014.

[44] J. Wu and X. Li, "Collective synchronization of kuramoto-oscillator networks," *IEEE Circuits and Systems Magazine*, vol. 20, no. 3, pp. 46–67, 2020.

[45] N. Chopra and M. W. Spong, "Passivity-based control of multi-agent systems," *Advances in robot control: from everyday physics to human-like movements*, pp. 107–134, 2006.

[46] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse problems*, vol. 34, no. 1, p. 014004, 2017.

[47] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic gradient descent," in *ICLR: International Conference on Learning Representations*, 2015, pp. 1–15.