2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Universal approximation of flows of control systems by recurrent neural networks

Miguel Aguiar[1] Amritam Das[2] Karl H. Johansson[1]

***Abstract*—We consider the problem of approximating flow functions of continuous-time dynamical systems with inputs. It is well-known that continuous-time recurrent neural networks are universal approximators of this type of system. In this paper, we prove that an architecture based on discrete-time recurrent neural networks universally approximates flows of continuous-time dynamical systems with inputs. The required assumptions are shown to hold for systems whose dynamics are well-behaved ordinary differential equations and with practically relevant classes of input signals. This enables the use of off-the-shelf solutions for learning such flow functions in continuous-time from sampled trajectory data.**

***Index Terms*—Machine learning, Neural networks, Nonlinear systems**

I. Introduction

The advantage of continuous-time models for learning dynamics has been pointed out in a number of recent works [1], [2], [3]. Such models naturally handle irregularly sampled or missing data, and are the natural model class for most physical systems.

Some approaches for continuous-time identification have been proposed [4], but for nonlinear systems the majority of research concentrates on discrete-time models [5]. A number of modeling approaches have arisen using ideas and model classes from classical and deep machine learning. It is well known that continuous-time recurrent neural networks can approximate large classes of continuous-time dynamical systems with inputs, see Sontag [6], Li et al [7] and references therein. In fact, these networks are able to approximate flows of stable continuous-time systems over unbounded time intervals [8], as well as more general input-output operators [9]. Neural Ordinary Differential Equations (Neural ODEs) [1] are a particular class of continuous-time models proposed to replace standard network layers appearing in models used for common learning tasks, and have been shown to be competitive with state-of-the-art models in system identification [10]. In [11] some specific architectures and learning methods for identifying differential equation models of control systems using neural networks are presented.

An assortment of related methods have been proposed for modeling autonomous systems with applications in the physical sciences [12], [13], [14]. For methods based on Koopman operator approximation in particular, extensions to certain classes of systems with inputs are possible [15]. Physics-informed learning has also emerged as a paradigm for learning solutions of ordinary and partial differential equations from data [16], [17]. These methods incorporate a set of differential equations known to be satisfied by the data as a regulariser in the loss function used to train the network, and can also be used to identify parameters in the equations.

In contrast to the majority of these approaches, the class of methods known as neural operator methods attempt to directly learn the solution operator of a differential equation, that is, the operator mapping initial conditions, forcing terms and parameters to the corresponding solution, rather than identifying the governing equations [18], [19], [20], [21]. The focus is then on engineering architectures with the appropriate inductive biases for a particular class of problems.

In this paper, we consider the problem of approximating the flow function of a dynamical system, that is, the solution operator mapping initial conditions and control inputs to the corresponding trajectory of the system. Exploiting the discrete structure of commonly used classes of control inputs, we show that the flow function can be exactly represented by a discrete-time dynamical system. This motivates the use of (discrete-time) Recurrent Neural Network (RNN) architectures to learn flow functions from data. We propose one such architecture which ensures that the trajectories of the learned model are continuous. In previous work [22], we have shown through numerical experiments that the architecture successfully learns flows of oscillators with complex dynamics, and have investigated its generalisation performance.

In this paper, our contribution is twofold. Firstly, we prove that the proposed architecture is a universal approximator for flow functions of control systems, which guarantees the well-posedness of the learning problem that we formulate mathematically in [22]. Secondly, we show by system-theoretic arguments that the required assumptions hold for systems whose dynamics are given by well-behaved ODEs, with rather general and practically relevant classes of input signals.

This approach has a number of advantages in comparison with methods based on learning the right-hand side of a differential equation. Errors in the learned dynamics can be propagated and affect long-term prediction performance. When the flow is directly approximated, the need for integration is obviated. This has the additional advantage of reducing the computational burden at both training and prediction time.

[1]Digital Futures and Division of Decision and Control Systems, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. Email: {aguiar, kallej}@kth.se, [2]Control Systems Group, EE Dept., Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. Email: am.das@tue.nl

In effect, under our formulation, the problem of learning a flow function amounts to a standard regression problem, and thus enables the use of off-the-shelf learning frameworks for training the model. At prediction time, one can query the solution map at any time instant, and, since the model uses standard neural network components, gradients of the flow with respect to, e.g., initial conditions or control values can be computed in a straightforward manner through automatic differentiation. Furthermore, the approach is able to accommodate more general classes of systems than those with dynamics given by ODEs.

The rest of the paper is organised as follows. Section II introduces notation and some basic definitions. In Section III we describe the proposed architecture and the considered class of input signals. This is followed by the statement of Theorem 1 in Section IV. In Section V we give a proof that discrete-time RNNs are universal approximators, an essential step in the proof of Theorem 1 given in Section VI. Section VII treats the case of flows of ODEs and the assumptions of Theorem 1 are shown to hold in that setting. A numerical example is briefly discussed in Section VIII, and concluding remarks are given in Section IX.

II. Preliminaries

A. Notation

The indicator function of a set A is written $\mathbf{1}_A$, and the identity function on A is written id_A. Sequences are written $(z_k)_{k=0}^{\infty}$, or in short-hand (z_k). The space of sequences with values in A is written $S(A) := \{(z_k)_{k=0}^{\infty} : z_k \in A\}$. The space of continuous functions $f : A \to \mathbb{R}^n$ on a compact set $A \subset \mathbb{R}^m$ is written $C_n(A)$. For $A \subset \mathbb{R}^n$ and $\varepsilon > 0$, $N_\varepsilon(A)$ denotes the (closed) ε-neighbourhood of A, i.e., the set of points at most ε distance away from A. If A is compact, then so is $N_\varepsilon(A)$. Vectors $v \in \mathbb{R}^d$ are written $v = (v_1, \ldots, v_d)$. We denote by $\|\cdot\|$ the Euclidean norm on $\mathbb{R}^d$, and for a matrix $M \in \mathbb{R}^{m\times n}$, $\|M\|$ denotes the induced operator norm.

B. Flows of controls systems in continuous-time

In this paper we consider finite-dimensional time-invariant control systems in continuous-time with state evolving in an open set $X \subset \mathbb{R}^{d_x}$. Such systems can be described abstractly by a *flow function*

$$\varphi : \mathbb{R}_{\geq 0} \times X \times \mathbb{U} \to X \tag{1}$$

where $\mathbb{U}$ is a given set of (control) inputs $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u}$. The flow satisfies the following properties (cf. Sontag [23, Chapter 2]):

- Identity: $\varphi(0, x, u) = x$
- Semigroup: $\varphi(s + t, x, u) = \varphi(t, \varphi(s, x, u), u^s)$

for all $x \in X$, $u \in \mathbb{U}$ and $s, t \geq 0$. Here $u^s \in \mathbb{U}$ denotes the input u shifted by $s > 0$ time units, i.e., $u^s(t) := u(t+s)$. The function $t \mapsto \varphi(t, x, u), t \geq 0$ is the trajectory of the system with initial state x when the applied control is u.

C. Neural networks as function approximators

In this paper, a *(feedforward) neural network* is any function $h : \mathbb{R}^m \to \mathbb{R}^n$ which can be written as

$$h(x) = C\sigma_p(Ax + b) + d, \ x \in \mathbb{R}^m \tag{2}$$

for $A \in \mathbb{R}^{p\times m}$, $b \in \mathbb{R}^p$, $C \in \mathbb{R}^{n\times p}$, $d \in \mathbb{R}^n$. Here $\sigma_p : \mathbb{R}^p \to \mathbb{R}^p$ is a diagonal mapping such that the *activation function* $\sigma : \mathbb{R} \to \mathbb{R}$ is applied to each coordinate, i.e., $\sigma_p(v) = (\sigma(v_1), \ldots, \sigma(v_p))$ for $v \in \mathbb{R}^p$. In practical applications, these are usually known as networks with *one hidden layer*.

We let $\mathfrak{N}_{\sigma,p}^{m,n}$ be the class of such networks and define $\mathfrak{N}_{\sigma}^{m,n} := \cup_{p=1}^{\infty} \mathfrak{N}_{\sigma,p}^{m,n}$. Throughout the paper, we shall assume that $\sigma : \mathbb{R} \to \mathbb{R}$ is a fixed bounded, continuous and nonconstant function. Under this assumption it is well-known [24] that $\mathfrak{N}_{\sigma}^{n,m}$ is dense in $C_n(K)$ for any compact set $K \subset \mathbb{R}^m$. That is, for any continuous function $f : K \to \mathbb{R}^n$ and $\varepsilon > 0$ there is a network $h \in \mathfrak{N}_{\sigma}^{n,m}$ such that $\sup_{x\in K} \|f(x) - h(x)\| < \varepsilon$.

Let $\mathfrak{N}_{\sigma,p}^{0} \subset \cup_{q\geq p} \mathfrak{N}_{\sigma,p}^{q,p}$ be the class of feedforward networks for which $C = I$ and $d = 0$ in (2). A *Recurrent Neural Network* (RNN) is then simply a difference equation whose right-hand side is a network in $\mathfrak{N}_{\sigma,p}^{0}$ for some $p \geq 0$:

Definition 1 (RNN)**.** An RNN is a difference equation of the form

$$z_{k+1} = \sigma_{d_z}(Az_k + Bu_k + b), \ k \in \mathbb{Z}_{\geq 0},$$

where $z \in \mathbb{R}^{d_z}$, $u \in \mathbb{R}^{d_u}$.

III. Architecture Definition

In this section we define a discrete-time RNN-based architecture to approximate flow functions of continuous-time dynamical systems. We focus in particular on systems for which the trajectories $t \mapsto \varphi(t, x, u)$ are continuous in time t. This is the case when φ arises from a differential equation, differential algebraic equation, but excludes e.g. hybrid systems with state jumps. We shall show that φ can be approximated by a function $\hat{\varphi}$ on a finite time interval, where $\hat{\varphi}(t, x, u)$ is computed by an RNN. In the following sections we make this precise.

A. Class of inputs

In order to approximate φ, we must impose some structure on $\mathbb{U}$. In practice, the majority of systems are controlled by a computer with a zero-order hold digital-to-analog converter, so that the input signal will be piecewise constant, with the control value changing at regular time instants with some period $\Delta > 0$. Occasionally, first- or higher-order polynomial parameterisations are also used. In this paper we consider a general parameterisation of control inputs which encompasses all of these cases. Namely, we assume that the control can be parameterised by a sequence of finite-dimensional parameters $(\omega_k)_{k=0}^{\infty} \subset \mathbb{R}^{d_\omega}$ as follows:

$$u(t) = \sum_{k=0}^{\infty} \alpha\left(\omega_k, \frac{t}{\Delta}\right) \mathbf{1}_{[k\Delta,(k+1)\Delta)}(t), \ t \geq 0. \tag{3}$$

Here $\alpha : \mathbb{R}^{d_\omega} \times \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u}$ is periodic with period 1 in its second argument. In other words, we have for each $k \geq 0$

$$u(t) = \alpha(\omega_k, t/\Delta),\ k\Delta \leq t < (k+1)\Delta.$$

The simplest example is given by $\alpha(\omega, t) := \omega$, corresponding to the case of piecewise constant controls with period Δ.

Throughout the paper we assume that Δ and the function α are fixed and known. For a set $\Omega \subset \mathbb{R}^{d_\omega}$ we define the set $\mathbb{U}(\Omega)$ of controls u parameterised by sequences in $S(\Omega)$ according to (3), i.e.,

$$\mathbb{U}(\Omega) := \Big\{ u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u} : (\omega_k)_{k=0}^{\infty} \in S(\Omega), \quad u(t) = \sum_{k=0}^{\infty} \alpha\left(\omega_k, \frac{t}{\Delta}\right) \mathbf{1}_{[k\Delta,(k+1)\Delta)}(t) \Big\}. \tag{4}$$

B. Representing flows by discrete-time systems

We let u_ω be the control generated by the constant sequence with value ω, so that $u_\omega(t) = \alpha(\omega, t/\Delta),\ t \geq 0$ and define the function $\Phi : [0,1] \times X \times \mathbb{R}^{d_\omega} \to X$ by

$$\Phi(\tau, x, \omega) := \varphi(\tau\Delta, x, u_\omega).$$

Fix $t \in \mathbb{R}_{\geq 0}$ and define $k_t := \lfloor t/\Delta \rfloor,\ \tau_t := (t - k_t\Delta)/\Delta$. The value of $\varphi(t, x, u)$ can be computed recursively by Φ as follows:

$$\begin{aligned} x_0 &= x \\ x_{k+1} &= \Phi(1, x_k, \omega_k),\ 0 \leq k < k_t \\ x_{k_t+1} &= \Phi(\tau_t, x_{k_t}, \omega_{k_t}) = \varphi(t, x, u). \end{aligned} \tag{5}$$

This can be seen as representing φ by a discrete-time system with inputs $(\tau, \omega) \in [0,1] \times \mathbb{R}^{d_\omega}$. Note that such a representation does not amount to a *discretisation* of φ, so that no loss of information or generality is incurred, and we are able to compute the flow φ at any instant of time through this correspondence.

The discrete-time system defined by (5) can be approximated by an RNN as follows. Let $x \in X$, $t \geq 0$ and $u \in \mathbb{U}$ be parameterised according to (3) by a sequence (ω_k). Fixing networks $h \in \mathfrak{N}^0_{\sigma,d_z}$, $\beta \in \mathfrak{N}^{d_x,d_z}_\sigma$ and $\gamma \in \mathfrak{N}^{d_z,d_x}_\sigma$, compute the sequence

$$\begin{aligned} z_0 &= \beta(x) \\ z_{k+1} &= h(1, z_k, \omega_k),\ 0 \leq k < k_t \\ z_{k_t+1} &= h(\tau_t, z_{k_t}, \omega_{k_t}) \end{aligned} \tag{6}$$

and set

$$\hat{\varphi}(t, x, u) = \gamma((1 - \tau_t) z_{k_t} + \tau_t z_{k_t+1}).$$

The interpolation guarantees that $\hat{\varphi}$ is continuous in t. Note that it does not amount to a linear interpolation, as z_{k_t+1} depends on τ_t.

In order to express $\hat{\varphi}$ explicitly, the following definition is useful, and will be used throughout the following sections.

Definition 2 (Recursion Map)**.** Let $f : A \times B \to A$. The associated *recursion map* $\rho_f : \mathbb{Z}_{\geq 0} \times A \times S(B) \to A$ is defined as

$$\begin{aligned} \rho_f(0, x, (u_k)) &= x, \\ \rho_f(n+1, x, (u_k)) &= f(\rho_f(n, x, (u_k)), u_n),\ n \geq 0. \end{aligned} \tag{7}$$

Now let $(\mathsf{t}^t_k)_{k=0}^{\infty} \in S([0,1])$ be defined by

$$\mathsf{t}^t_k = \begin{cases} 1, & 0 \leq k < k_t \\ \tau_t, & k = k_t \\ 0, & k > k_t. \end{cases} \tag{8}$$

Then we can rewrite (6) as[1] $z_k = \rho_h(k, \beta(x), (\mathsf{t}^t_k, \omega_k))$, $k \geq 0$, and thus $\hat{\varphi}$ can be written

$$\begin{aligned} \hat{\varphi}(t, x, u) = \gamma[&(1 - \tau_t)\rho_h(k_t, \beta(x), (\mathsf{t}^t_k, \omega_k)) \\ &+ \tau_t \rho_h(k_t + 1, \beta(x), (\mathsf{t}^t_k, \omega_k))]. \end{aligned}$$

We let $\mathcal{H}$ denote the set of functions $\hat{\varphi} : \mathbb{R}_{\geq 0} \times X \times \mathbb{U} \to \mathbb{R}^{d_x}$ defined in this way, that is,

$$\begin{aligned} \mathcal{H} := \Big\{ &\hat{\varphi} : \mathbb{R}_{\geq 0} \times X \times \mathbb{U} \to \mathbb{R}^{d_x} : d_z \in \mathbb{Z}_{\geq 0}, \\ &\gamma \in \mathfrak{N}^{d_z,d_x}_\sigma,\ h \in \mathfrak{N}^0_{\sigma,d_z},\ \beta \in \mathfrak{N}^{d_x,d_z}_\sigma, \\ &\hat{\varphi}(t, x, u) = \gamma[(1 - \tau_t)\rho_h(k_t, \beta(x), (\mathsf{t}^t_k, \omega_k)) \\ &\qquad + \tau_t \rho_h(k_t + 1, \beta(x), (\mathsf{t}^t_k, \omega_k))] \Big\}. \end{aligned} \tag{9}$$

For a more detailed explanation and motivation of the architecture just described, the reader is referred to [22].

IV. Statement of the Main Result

Theorem 1. *Suppose the flow of a control system $\varphi : \mathbb{R}_{\geq 0} \times X \times \mathbb{U} \to X$ satisfies the following conditions:*

1) *Given a compact set $K_\omega \subset \mathbb{R}^{d_\omega}$, define $\mathbb{U}(K_\omega)$ according to* (4)*. Then $\mathbb{U}(K_\omega) \subset \mathbb{U}$, i.e., for any $u \in \mathbb{U}(K_\omega)$, the corresponding trajectory $\varphi(\cdot, x, u)$ is well-defined for all $x \in X$.*
2) *The function $\Phi : [0,1] \times X \times \mathbb{R}^{d_\omega} \to X$ defined as*

$$\Phi(\tau, x, \omega) := \varphi(\tau\Delta, x, u_\omega),\ u_\omega(t) = \alpha(\omega, t/\Delta)$$

is right-differentiable at $\tau = 0$ for every $(x, \omega) \in X \times \mathbb{R}^{d_\omega}$.

3) *The function $\Psi : [0,1] \times X \times \mathbb{R}^{d_\omega}$ defined as*

$$\Psi(\tau, x, \omega) := \begin{cases} x + \tau^{-1}(\Phi(\tau, x, \omega) - x),\ \tau \in (0,1] \\ \lim\limits_{t \downarrow 0} \left[x + t^{-1}(\Phi(t, x, \omega) - x)\right],\ \tau = 0 \end{cases} \tag{10}$$

is continuous and locally Lipschitz in x.

Then, for any $\varepsilon > 0$, $T \geq 0$ and compact sets $K_x \subset X$, $K_\omega \subset \mathbb{R}^{d_\omega}$, there exists $\hat{\varphi} \in \mathcal{H}$, defined according to (9)*, such that $\|\varphi(t, x, u) - \hat{\varphi}(t, x, u)\| < \varepsilon$ holds for all $t \in [0, T]$, $x \in K_x$ and $u \in \mathbb{U}(K_\omega)$. Furthermore, γ and β in* (9) *can be chosen to be affine with $\gamma \circ \beta = \mathrm{id}_{\mathbb{R}^{d_x}}$.*

[1] With some abuse of notation, we interpret h as a function mapping $\mathbb{R}^{d_z} \times ([0,1] \times \mathbb{R}^{d_\omega})$ to $\mathbb{R}^{d_z}$.

Note that assumptions 2 and 3 implicitly represent assumptions on φ and α. In Section VII we shall give conditions under which they are satisfied for flows of differential equations.

V. Universal Approximation of Discrete-Time Systems

In this section we give a proof that discrete-time RNNs are universal approximators of discrete-time systems, a fact that will be used in the proof of Theorem 1.

Theorem 2 (Universal approximation for discrete-time dynamical systems). *Let $f : \mathbb{R}^{d_x} \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_x}$ be a continuous function that is locally Lipschitz in the first variable, in the sense that for any compact set $K \subset \mathbb{R}^{d_x}$ there exists a locally bounded function $\nu_K : \mathbb{R}^{d_u} \to \mathbb{R}_{\geq 0}$ such that*

$$\|f(x_2, u) - f(x_1, u)\| \leq \nu_K(u) \|x_2 - x_1\|, \ x_1, x_2 \in K.$$

Then for any $\varepsilon > 0$, $N \in \mathbb{Z}_{\geq 0}$ and compact sets $K_x \subset \mathbb{R}^{d_x}$ and $K_u \subset \mathbb{R}^{d_u}$ there exist networks $h \in \mathfrak{N}^0_{\sigma,d_z}$, $\gamma \in \mathfrak{N}^{d_z,d_x}_\sigma$ and $\beta \in \mathfrak{N}^{d_x,d_z}_\sigma$ such that for any $x \in K_x$ and $u \in S(K_u)$ we have

$$\|\rho_f(n, x, u) - \gamma(\rho_h(n, \beta(x), u))\| < \varepsilon, \ n = 0, \dots, N, \quad (11)$$

where ρ_f is defined as in (7). Furthermore, γ and β can be chosen to be affine with $\gamma \circ \beta = \mathrm{id}_{\mathbb{R}^{d_x}}$.

Noe that h above defines an RNN. Therefore, Theorem 2 states that any discrete-time dynamical system with a locally Lipschitz right-hand side can be approximated in the sense of (11) by an RNN. This is a well-known fact [6], [25], but for reference we include a full proof under the stated assumptions, as this result is an important step in the proof of Theorem 1, and there we will in particular use the fact that $\gamma \circ \beta = \mathrm{id}_{\mathbb{R}^{d_x}}$ and that these maps are affine.

Proof. The case $N = 0$ is trivial and $N = 1$ corresponds to the standard universal approximation theorem proved in Hornik [24], so we assume $N \geq 2$ in what follows.

Define the sets $K^0_x, \dots, K^{N-1}_x$ recursively by $K^{n+1}_x = f(K^n_x, K_u)$ with $K^0_x = K_x$. By continuity of f, the K^n_x are compact. For any input sequence $u \in S(K_u)$ and initial state $x_0 \in K_x$, we then have that

$$\rho_f(n, x_0, u) \in K^n_x, \ n = 0, 1, \dots, N - 1.$$

Define also $L^n_f, \eta^n_f \geq 0$ and sets $\tilde{K}^n$, $n = 1, \dots, N - 1$ recursively as follows:

$$\eta^1_f = 1$$
$$\tilde{K}^n = N_{\varepsilon\eta^n_f}(K^n_x), \ L^n_f = \max\left\{1, \sup_{u \in K_u} \nu_{\tilde{K}^n}(u)\right\}$$
$$\eta^{n+1}_f = 1 + L^n_f \eta^n_f,$$

and let

$$K = K_x \cup \bigcup_{n=1}^{N-1} \tilde{K}^n,$$
$$\varepsilon_n = \frac{1}{2^{N-n} \prod_{k=n}^{N-1} L^k_f} \varepsilon, \ n = 1, \dots, N.$$

Pick a neural network $g \in \mathfrak{N}^{d_x+d_u,d_x}_\sigma$ such that

$$\sup_{x \in K, u \in K_u} \|f(x, u) - g(x, u)\| < \min_{n=1,\dots,N} \varepsilon_n. \quad (12)$$

In particular, $\sup_{x \in K, u \in K_u} \|f(x, u) - g(x, u)\| < \varepsilon$.

Now, pick $x \in K_x$ and $u \in S(K_u)$. We have (omitting the (x, u) arguments since they are fixed everywhere)

$$\begin{aligned}
&\|\rho_f(n + 1, x, u) - \rho_g(n + 1, x, u)\| \\
&\quad = \|f(\rho_f(n)) - g(\rho_g(n))\| \\
&\quad \leq \|f(\rho_f(n)) - f(\rho_g(n))\| + \|f(\rho_g(n)) - g(\rho_g(n))\|
\end{aligned}$$

Assuming that $\|\rho_f(n) - \rho_g(n)\| < \varepsilon\eta^n_f$, we have $\rho_g(n) \in \tilde{K}^n \subset K$ and so

$$\begin{aligned}
\|\rho_f(n + 1) - \rho_g(n + 1)\| &< L^n_f \|\rho_f(n) - \rho_g(n)\| + \varepsilon \\
&\leq \varepsilon\eta^{n+1}_f.
\end{aligned}$$

Since (12) implies

$$\|\rho_f(1, x, u) - \rho_g(1, x, u)\| < \varepsilon \, (= \varepsilon\eta^1_f),$$

by induction we have that $\|\rho_f(n, x, u) - \rho_g(n, x, u)\| < \varepsilon\eta^n_f$ for $n = 1, \dots, N - 1$, so that $\rho_g(n, x, u) \in \tilde{K}^n$.

Now, we show by induction that

$$\|\rho_f(n, x, u) - \rho_g(n, x, u)\| < \varepsilon_n$$

for each $n \geq 0$. For $n = 1$ this holds by (12):

$$\|\rho_f(1, x, u) - \rho_g(1, x, u)\| = \|f(x, u_0) - g(x, u_0)\| < \varepsilon_1.$$

Assume that $\|\rho_f(n, x, u) - \rho_g(n, x, u)\| < \varepsilon_n$. Then

$$\begin{aligned}
&\|\rho_f(n + 1, x, u) - \rho_g(n + 1, x, u)\| \\
&\quad \leq \|f(\rho_f(n)) - f(\rho_g(n))\| + \|f(\rho_g(n)) - g(\rho_g(n))\| \\
&\quad < L^n_f \varepsilon_n + \varepsilon_n = \frac{\varepsilon_{n+1}}{2} + \varepsilon_n \leq \varepsilon_{n+1}.
\end{aligned}$$

And since $\varepsilon_n \leq \varepsilon$ for $n = 1, \dots, N$, we have that $\|\rho_f(n, x, u) - \rho_g(n, x, u)\| < \varepsilon$, as desired.

Finally, since it is not necessarily the case that $g \in \mathfrak{N}^0_{\sigma,p}$ for some p, it remains to obtain an equivalent recurrent neural network. Write g explicitly as

$$g(x, u) = T\sigma_p(Ax + Bu + b) + c,$$

and rank-factorise T as

$$T = M \begin{bmatrix} T_1 \\ 0 \end{bmatrix}$$

with $M \in \mathbb{R}^{d_x \times d_x}$ invertible and $T_1 \in \mathbb{R}^{r \times p}$ of full row rank. Then, with

$$g_1(x, u) := \begin{bmatrix} T_1\sigma_p(AMx + Bu + b) + c'_1 \\ c'_2 \end{bmatrix}, \ M^{-1}c = \begin{bmatrix} c'_1 \\ c'_2 \end{bmatrix},$$

it follows that

$$M\rho_{g_1}(n, M^{-1}x, u) = \rho_g(n, x, u)$$

for all (n, x, u). Let now T^+_1 be a right inverse of T_1 (i.e., $T_1T^+_1 = I_r$) and

$$Q := M \begin{bmatrix} T_1 & 0 \\ 0 & I_{d_x-r} \end{bmatrix}, \ Q^+ := \begin{bmatrix} T^+_1 & 0 \\ 0 & I_{d_x-r} \end{bmatrix} M^{-1}.$$

Then with

$$g_2(z,u) := \begin{bmatrix} \sigma_p(AQz + Bu + b) + T_1^+ c_1' \\ c_2' \end{bmatrix},$$

we get

$$Q\rho_{g_2}(n, Q^+ x, u) = \rho_g(n, x, u).$$

Finally, let

$$\tilde{A} := \begin{bmatrix} AQ \\ 0_{(d_x - r)\times(p+d_x-r)} \end{bmatrix},\ \tilde{B} := \begin{bmatrix} B \\ 0_{(d_x-r)\times d_u} \end{bmatrix},$$

$$\tilde{b} := \begin{bmatrix} b \\ 0_{d_x - r} \end{bmatrix},\ \tilde{c} := \begin{bmatrix} T_1^+ c_1' \\ c_2' - \sigma_{d_x - r}(0) \end{bmatrix}$$

and define the maps $\gamma : \mathbb{R}^{p+d_x-r} \to \mathbb{R}^{d_x}$ and $\beta : \mathbb{R}^{d_x} \to \mathbb{R}^{p+d_x-r}$ as

$$\gamma(z) := Q(z + \tilde{c}),\ \beta(x) := Q^+ x - \tilde{c}.$$

Then with $d_z := p + d_x - r$ and

$$h(z,u) := \sigma_{d_z}(\tilde{A}z + \tilde{B}u + \tilde{b} + \tilde{A}\tilde{c})$$

we get

$$\gamma(\rho_h(n, \beta(x), u)) = \rho_g(n, x, u),$$

so that $h \in \mathfrak{N}^0_{\sigma, d_z}$ and

$$\|\gamma(\rho_h(n, \beta(x), u)) - \rho_f(n, x, u)\| < \varepsilon$$

for all $x \in K_x$, $u \in S(K_u)$ and $n = 0, \ldots, N$, as desired. Note also that γ and β have the desired properties. ■

VI. Proof of Theorem 1

We begin with some intuition on the definition of Ψ in (10) and the stated assumptions. Let $\hat{\varphi} \in \mathcal{H}$ and let β, h, γ be the corresponding networks as in (9). Assume for the moment that $\gamma = \beta = \mathrm{id}_{\mathbb{R}^{d_x}}$. In the first control period, i.e., for $0 < \tau \le 1$ it holds that

$$\begin{aligned}
&\varphi(\tau\Delta, x, u_\omega) - \hat{\varphi}(\tau\Delta, x, u_\omega) \\
&\quad = \Phi(\tau, x, \omega) - [(1-\tau)x + \tau h(\tau, x, \omega)] \\
&\quad = \tau\big(h(\tau, x, \omega) - \big[x + \tau^{-1}(\Phi(\tau, x, \omega) - x)\big]\big) \\
&\quad = \tau(h(\tau, x, \omega) - \Psi(\tau, x, \omega)).
\end{aligned}$$

Furthermore, note that $\Phi(1, x, \omega) = \Psi(1, x, \omega)$, so if we replace Φ by Ψ in (5) we get the same result, provided we interpolate the final state, that is,

$$\varphi(t, x, u) = \Phi(\tau_t, x_{k_t}, \omega_{k_t}) = (1-\tau_t)x_{k_t} + \tau_t \Psi(\tau_t, x_{k_t}, \omega_{k_t}).$$

This motivates the idea that we should approximate the discrete dynamical system obtained by iterating Ψ:

$$x_{k+1} = \Psi(\tau_k, x_k, \omega_k),\ k \ge 0.$$

Using the recursion map notation, we can equivalently write

$$x_k = \rho_\Psi(k, x_0, (\tau_k, \omega_k)_{k=0}^\infty).$$

By Theorem 2, there exists a network $h \in \mathfrak{N}^0_{\sigma, d_z}$ and affine maps γ, β such that

$$\|\gamma(\rho_h(n, \beta(x), (\tau_k, \omega_k))) - \rho_\Psi(n, x, (\tau_k, \omega_k))\| < \varepsilon \quad (13)$$

for $n = 0, \ldots, k_T + 1$ and any $x \in K_x$ and $(\tau_k, \omega_k) \in S([0,1] \times K_\omega)$. Let $\hat{\varphi} \in \mathcal{H}$ be defined by these three networks according to (9), and recall that $\gamma \circ \beta = \mathrm{id}_{\mathbb{R}^{d_x}}$.

Fix $x \in K_x$, $u \in \mathbb{U}(K_\omega)$ and $t \in [0, T]$. Let (ω_k) be a sequence parameterising the control u and define

$$\begin{aligned}
z_0 &= \beta(x) \\
z_{k+1} &= h(1, z_k, \omega_k),\ 0 \le k < k_t \\
z_{k_t+1} &= h(\tau_t, z_{k_t}, \omega_{k_t}).
\end{aligned}$$

Then, as before

$$z_n = \rho_h(n, \beta(x), (\mathfrak{t}_k^t, \omega_k)),\ 0 \le n \le k_t + 1$$

(recall the definition of $(\mathfrak{t}_k^t)$ in (8)). It follows from (13) that

$$\|\varphi(k\Delta, x, u) - \gamma(z_k)\| < \varepsilon$$

for $k = 0, \ldots, k_t$ and with $x_{k_t} := \varphi(k_t\Delta, x, u)$

$$\|\Psi(\tau_t, x_{k_t}, \omega_{k_t}) - \gamma(z_{k_t+1})\| < \varepsilon.$$

Write

$$\begin{aligned}
&\varphi(t, x, u) - \hat{\varphi}(t, x, u) \\
&\quad = \varphi(t, x, u) - \gamma((1-\tau_t)z_{k_t} + \tau_t z_{k_t+1}) \\
&\quad = \Phi(\tau_t, x_{k_t}, \omega_{k_t}) - \gamma((1-\tau_t)z_{k_t} + \tau_t z_{k_t+1}) \\
&\quad = x_{k_t} + \tau_t(\Psi(\tau_t, x_{k_t}, \omega_{k_t}) - x_{k_t}) \\
&\qquad - \gamma((1-\tau_t)z_{k_t} + \tau_t z_{k_t+1}) \\
&\quad = (1-\tau_t)x_{k_t} + \tau_t \Psi(\tau_t, x_{k_t}, \omega_{k_t}) \\
&\qquad - (1-\tau_t)\gamma(z_{k_t}) - \tau_t \gamma(z_{k_t+1}) \\
&\quad = (1-\tau_t)(x_{k_t} - \gamma(z_{k_t})) + \tau_t(\Psi(\tau_t, x_{k_t}, \omega_{k_t}) - \gamma(z_{k_t+1})).
\end{aligned}$$

If $t < \Delta$ then $k_t = 0$, so that

$$\begin{aligned}
&\varphi(t, x, u) - \hat{\varphi}(t, x, u) \\
&\quad = (1-\tau_t)(x - \gamma(z_0)) + \tau_t(\Psi(\tau_t, x, \omega_0) - \gamma(z_1)) \\
&\quad = (1-\tau_t)(x - \gamma(\beta(x))) + \tau_t(\Psi(\tau_t, x, \omega_0) - \gamma(z_1)) \\
&\quad = \tau_t(\Psi(\tau_t, x, \omega_0) - \gamma(z_1)),
\end{aligned}$$

and thus

$$\begin{aligned}
\|\varphi(t, x, u) - \hat{\varphi}(t, x, u)\| &= \tau_t \|\Psi(\tau_t, x, \omega_0) - \gamma(z_0)\| \\
&< \varepsilon\tau_t \le \varepsilon.
\end{aligned}$$

For $t \ge \Delta$, we have

$$\begin{aligned}
\|\varphi(t, x, u) - \hat{\varphi}(t, x, u)\| &\le (1-\tau_t)\|x_{k_t} - \gamma(z_{k_t})\| \\
&\quad + \tau_t \|\Psi(\tau_t, x_{k_t}, \omega_{k_t}) - \gamma(z_{k_t+1})\| \\
&< \varepsilon,
\end{aligned}$$

and the proof is complete.

VII. Flows of Differential Equations

In this section, we consider the class of flows φ arising from a controlled Ordinary Differential Equation (ODE) of the form

$$\dot{\xi}(t) = f(\xi(t), u(t)),\ \xi(0) = x. \quad (14)$$

If the function $f : X \times \mathbb{R}^{d_u} \to \mathbb{R}^{d_x}$ is sufficiently regular, the flow of such a system is well-defined for all Borel measurable and essentially bounded controls [23], and satisfies the ODE in the following sense:

$$\varphi(t,x,u) = x + \int_0^t f(\varphi(s,x,u),u(s))\mathrm{d}s,\ t \in \mathbb{R}_{\geq 0}. \quad (15)$$

In particular, if f is continuous and the control u is right-continuous at time $s \geq 0$, then it holds that

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{t=s} \varphi(t,x,u) = f(\varphi(s,x,u),u(s)). \quad (16)$$

We now show that the assumptions in Theorem 1 are satisfied in this case under mild conditions on the input parameterisation α and the right-hand side f of the ODE.

Lemma 1. *Assume that the functions f in* (14) *in and α in* (3) *satisfy the following conditions*

I) The function α is measurable, and for each $\omega \in \mathbb{R}^{d_\omega}$ the function $\alpha(\omega,\cdot)$ is bounded on $[0,1]$ and right-continuous at $t=0$. Furthermore, the family of functions $\{\alpha(\cdot,t) : \mathbb{R}^{d_\omega} \to \mathbb{R}^{d_u} : t \in [0,1)\}$ is equicontinuous, i.e., if $\omega_n \to \omega$ then for any $\varepsilon > 0$ there exists $N \in \mathbb{Z}_{\geq 0}$ such that for all $t \in [0,1)$ and $n \geq N$ it holds that $\|\alpha(\omega_n,t) - \alpha(\omega,t)\| < \varepsilon$.

II) The function f is continuously differentiable in (x,u), and solutions to (14) *exist in the sense of* (15) *for $t \in \mathbb{R}_{\geq 0}$, for all $x \in X$ and all measurable and essentially bounded controls $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u}$.*

Then the flow φ associated to the ODE (14) *satisfies the assumptions of Theorem 1.*

Recalling the definition of u_ω in Theorem 1, Assumption I implies that u_ω is continuous from the right at $t=0$ and that $u_{\omega_n} \to u_\omega$ uniformly when $\omega_n \to \omega$.

Assumption II above is sometimes referred to as *forward completeness* of (14). There is no single condition on f that can guarantee forward completeness; examples of possible conditions are discussed in [23], [26]. It implies the existence of φ satisfying (15) for all $t \geq 0$, and that $\mathbb{U}$ can be chosen to be the set of all measurable essentially bounded functions $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^{d_u}$.

Proof. First, we show that Assumption 1 in Theorem 1 holds. Let $K_\omega \subset \mathbb{R}^{d_\omega}$ be a compact set, and let $u \in \mathbb{U}(K_\omega)$ be parameterised by the sequence (ω_k). Let $a_k := \sup_{t\in[0,1]} \|\alpha(\omega_k,t)\|$. Suppose (a_k) is unbounded, and pick a subsequence (a_{k_j}) such that $\lim_{j\to\infty} a_{k_j} = \infty$. By compactness, there is a subsequence $(\omega_{k'_j})$ of (ω_{k_j}) with $\lim_{j\to\infty} \omega_{k'_j} = \omega \in K_\omega$. Then, by equicontinuity we have that for j large enough

$$\left\|\alpha(\omega_{k'_j},t)\right\| < 1 + \|\alpha(\omega,t)\|,\ t \in [0,1),$$

which implies that $(a_{k'_j})$ is bounded, a contradiction. Hence a_k is bounded, and so u is bounded. Since α is measurable, so is u, and thus $u \in \mathbb{U}$, as desired.

We now show that Assumption 2 in Theorem 1 holds. Since α is right-continuous at $t=0$, (16) gives

$$\begin{aligned}\left.\frac{\mathrm{d}}{\mathrm{d}\tau}\right|_{\tau=0} \Phi(\tau,x,\omega) &= \left.\frac{\mathrm{d}}{\mathrm{d}\tau}\right|_{\tau=0} \varphi(\Delta\tau,x,u_\omega) \\ &= f(\varphi(0,x,u_\omega),u_\omega(0))\Delta = f(x,\alpha(\omega,0))\Delta,\end{aligned}$$

and hence Φ is differentiable from the right at $\tau = 0$, as desired.

Finally, we show that Assumption 3 in Theorem 1 holds. Let $\Psi_0(\tau,x,\omega) = \Psi(\tau,x,\omega) - x$. We will show that the differential of Ψ_0 with respect to x is continuous, and thus bounded on compact sets, from which it follows that Ψ_0 (and thus Ψ) is locally Lipschitz. The remainder of the proof requires a few additional properties of the flow φ which we state in the following sublemmata.

Sublemma 1. *Let $(\omega_n) \subset \mathbb{R}^{d_\omega}$ and $(x_n) \subset X$ be such that $\omega_n \to \omega$ and $x_n \to x \in X$. Then $\varphi(t,x_n,u_{\omega_n}) \to \varphi(t,x,u_\omega)$ uniformly in $t \in [0,\Delta]$.*

Proof. See Sontag [23], Theorem 1. ∎

Sublemma 2. *The flow φ is differentiable with respect to the initial condition x and and its differential with respect to x, $D_x\varphi$, satisfies*

$$D_x\varphi(t,x,u)\xi = \lambda_{x,u}(t;\xi)$$

for all $\xi \in \mathbb{R}^n$, where $\lambda_{x,u}$ is the solution of the linear boundary value problem

$$\begin{aligned}\dot{\lambda}_{x,u}(s;\xi) &= D_x f(\varphi(s,x,u),u(s))\lambda(s;\xi) \\ \lambda_{x,u}(0;\xi) &= \xi.\end{aligned} \quad (17)$$

Equivalently, $D_x\varphi(t,x,u) = \Lambda_{x,u}(t)$ where $\Lambda_{x,u}$ is the state transition matrix associated to the linear system (17).

Proof. See Sontag [23], Theorem 1. ∎

We also need the following result on the continuity of solutions of linear ODEs with respect to the coefficient matrix.

Sublemma 3. *Let $x, z : [t_1,t_2] \to \mathbb{R}^{d_x}$ satisfy*

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) \\ \dot{z}(t) &= B(t)z(t)\end{aligned} \qquad t \in (t_1,t_2)$$

with $A, B : [t_1,t_2] \to \mathbb{R}^{d_x \times d_x}$ measurable and essentially bounded. Then, with $d(t) = x(t) - z(t)$ and $D(t) := A(t) - B(t)$,

$$\|d(t)\| \leq \left(\|d(t_1)\| + \int_{t_1}^t \|D(s)\|\,\|x(s)\|\,\mathrm{d}s\right)\mathrm{e}^{\int_{t_1}^t \|B(s)\|\mathrm{d}s} \quad (18)$$

for $t \in [t_1,t_2]$.

Proof. Write

$$d(t) = d(t_1) + \int_{t_1}^t \left[B(s)d(s) + (A(s) - B(s))x(s)\right]\mathrm{d}s,$$

so that

$$\|d(t)\| \leq \|d(t_1)\| + \int_{t_1}^t \|B(s)\|\,\|d(s)\|\,\mathrm{d}s + \int_{t_1}^t \|D(s)\|\,\|x(s)\|\,\mathrm{d}s,$$

and Grönwall's inequality gives the desired result. ∎

We shall use the bound (18) to show that Λ_{x,u_ω} is continuous with respect to (x,ω). To this end, let $x_n \to x$ and $\omega_n \to \omega$ and, for $t \in [0,\Delta]$, set

$$A(t) = D_x f(\varphi(t,x,u_\omega), u_\omega(t)), \tag{19}$$
$$B_n(t) = D_x f(\varphi(t,x_n,u_{\omega_n}), u_{\omega_n}(t)). \tag{20}$$

By continuity of φ and α, there exist compact sets K'_x and K'_u such that $\varphi(t,x,u_\omega), \varphi(t,x_n,u_{\omega_n}) \in K'_x$ and $u_\omega(t), u_{\omega_n}(t) \in K'_u$ holds for all n and $t \in [0,\Delta]$. Let

$$\bar{F} := \sup\{\|D_x f(z,u)\| : \ z \in K'_x, \ u \in K'_u\}. \tag{21}$$

We have $\bar{F} < \infty$, by continuity of $D_x f$, and $\|B(s)\| \le \bar{F}$ for $s \in [0,\Delta]$. Sublemma 3 applied on the interval $[0,t]$ now gives

$$\begin{aligned}
&\left\|\lambda_{x,u_\omega}(t;\xi) - \lambda_{x_n,u_{\omega_n}}(t;\xi)\right\| \\
&\le \mathrm{e}^{t\bar{F}} \int_0^t \|A(s)-B_n(s)\| \, \|\lambda_{x,u_\omega}(s;\xi)\| \, \mathrm{d}s \\
&\le \mathrm{e}^{t\bar{F}} \sup_{s\in[0,\Delta]} \|\Lambda_{x,u_\omega}(s)\| \left(\int_0^t \|A(s)-B_n(s)\| \, \mathrm{d}s\right) \|\xi\|,
\end{aligned}$$

so that

$$\begin{aligned}
&\left\|\Lambda_{x,u_\omega}(t) - \Lambda_{x_n,u_{\omega_n}}(t)\right\| \\
&\quad \le \mathrm{e}^{t\bar{F}} \sup_{s\in[0,\Delta]} \|\Lambda_{x,u_\omega}(s)\| \int_0^t \|A(s)-B_n(s)\| \, \mathrm{d}s
\end{aligned} \tag{22}$$

for each t. By continuity of $D_x f$ and the uniform convergence of $\varphi(t,x_n,u_{\omega_n})$ and u_{ω_n}, B_n converges uniformly to A, and thus $\Lambda_{x_n,u_{\omega_n}} \to \Lambda_{x,u_\omega}$ uniformly on $[0,\Delta]$.

Returning to our original goal, for $\tau > 0$ we have

$$\begin{aligned}
D_x\Psi_0(\tau,x,\omega) &= \tau^{-1}(D_x\Phi(\tau,x,\omega) - I) \\
&= \tau^{-1}(D_x\varphi(\tau\Delta,x,u_\omega) - I) = \tau^{-1}(\Lambda_{x,u_\omega}(\tau\Delta) - I)
\end{aligned}$$

Due to the continuity of Λ_{x,u_ω}, $D_x\Psi_0$ is continuous in (τ,x,ω) for $\tau > 0$.

For $\tau = 0$ we have $D_x\Psi_0(0,x,\omega) = \Delta D_x f(x,u_\omega(0))$. If $\tau_n \to 0$ with $\tau_n > 0$, $x_n \to x$ and $\omega_n \to \omega$ then

$$\begin{aligned}
D_x\Psi_0(0,x,\omega) &- D_x\Psi_0(\tau_n,x_n,\omega) \\
&= \Delta D_x f(x,u_\omega(0)) - \tau_n^{-1}\big(\Lambda_{x_n,u_{\omega_n}}(\tau_n\Delta) - I\big) \\
&= \Delta D_x f(x,u_\omega(0)) - \tau_n^{-1}(\Lambda_{x,u_\omega}(\tau_n\Delta) - I) \\
&\quad + \tau_n^{-1}(\Lambda_{x,u_\omega}(\tau_n\Delta) - \Lambda_{x_n,u_{\omega_n}}(\tau_n\Delta)).
\end{aligned}$$

The first term of the last equality goes to zero, hence we are left to investigate the second term. With $A, B_n, \bar{F}$ defined in (19)-(21), from (22) we find

$$\begin{aligned}
&\left\|\tau_n^{-1}(\Lambda_{x,u_\omega}(\tau_n\Delta) - \Lambda_{x_n,u_{\omega_n}}(\tau_n\Delta))\right\| \\
&\le \mathrm{e}^{\tau_n\Delta\bar{F}} \sup_{t\in[0,\Delta]} \|\Lambda_{x,u_\omega}(t)\| \frac{1}{\tau_n}\int_0^{\tau_n\Delta} \|A(t)-B_n(t)\| \, \mathrm{d}t.
\end{aligned}$$

Pick $\varepsilon > 0$, and let n be large enough that $\|A(t) - B_n(t)\| < \varepsilon$ for $t \in [0,\Delta]$, so that

$$\tau_n^{-1}\left\|\Lambda_{x,u_\omega}(\tau_n\Delta) - \Lambda_{x_n,u_{\omega_n}}(\tau_n\Delta)\right\| \le \left(\Delta \mathrm{e}^{\tau_n\Delta\bar{F}} \sup_{t\in[0,\Delta]} \|\Lambda_{x,u_\omega}(t)\|\right)\varepsilon$$

and $\tau_n^{-1}(\Lambda_{x,u_\omega}(\tau_n\Delta) - \Lambda_{x_n,u_{\omega_n}}(\tau_n\Delta)) \to 0$ as $n \to \infty$, as desired.

■

VIII. Numerical Example

We illustrate the use of the proposed architecture for learning the flow map of a FitzHugh-Nagumo oscillator whose dynamics are given by

$$\begin{aligned}
\eta\dot{x}_1(t) &= x_1(t) - x_1(t)^3 - x_2(t) + u(t) \\
\eta\gamma\dot{x}_2(t) &= x_1(t) + a - bx_2(t),
\end{aligned} \tag{23}$$

where $\eta = 1/50$, $\gamma = 40$, $a = 0.3$, $b = 1.4$. The control u is piecewise constant (i.e. $\alpha(\omega,t) = \omega$) with $\Delta = 0.2$.

We generate data consisting of $N = 300$ trajectories of (23) on $t \in [0,20]$ with $x^n(0) \overset{\text{i.i.d.}}{\sim} N(0,I)$ and each input u^n is parameterised as in (3) by a sequence (ω_k) distributed as follows:

$$\begin{aligned}
\omega_{40k} &\overset{\text{i.i.d.}}{\sim} \text{LogNormal}(\mu = \log(0.2), \sigma = 0.5), \\
\omega_{j+40k} &= \omega_{40k}, \ j = 1,\ldots,39
\end{aligned}$$

for $k \in \mathbb{Z}_{\ge 0}$. In other words, the inputs u^n are square waves with a period of 8 time units and the amplitude at each period is sampled from a log-normal distribution. The dynamics (23) are integrated using a Backward Differentiation Formula method and for each trajectory $K = 300$ trajectory values $\xi_k^n = \varphi(t_k^n, x^n, u^n) + \varepsilon_k^n$ are sampled, where ε_k^n is Gaussian measurement noise with standard deviation equal to 0.05.

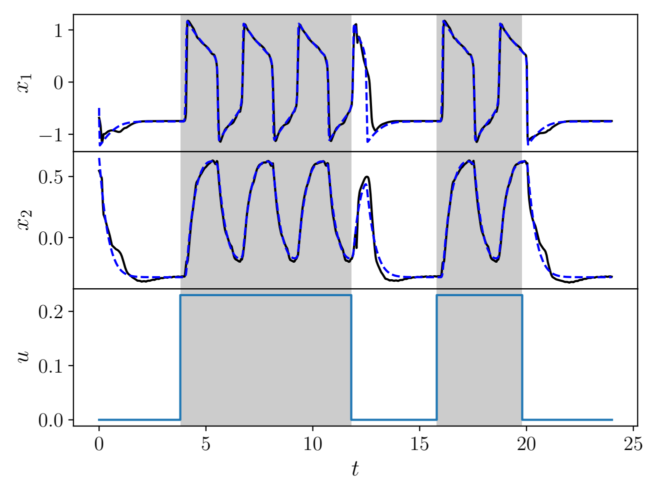

Fig. 1. Real (blue, dashed) and model (black) trajectories of the FitzHugh-Nagumo system (23) with the test input u.

We train a network defined according to our architecture, where h is a single-layer long short-term memory recurrent network with 32 hidden states, and γ, β are feedforward networks with 3 hidden layers with 32 nodes in each layer. The training is done by minimising the mean squared error loss

$$\frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\|\xi_k^n - \hat{\varphi}(t_k^n, x^n, u^n)\|^2 \tag{24}$$

using the `Adam` gradient descent algorithm. Due to space constraints we refer the reader to [22] for more details on the training procedure.

In Figure 1 the real state and predicted state trajectories for an input u and initial condition not in the training data set are plotted. The region shaded in grey indicates the times at which the applied input induces the excitable behaviour of the FitzHugh-Nagumo model [27], which is captured by the learned flow model.

IX. Conclusions

We have shown that the RNN-based architecture described in this paper is a universal approximator of flow functions of dynamical systems with control inputs. The required assumptions were shown to hold in the important case of flows of control systems described by ODEs. The parameterisation of control inputs, from which the discrete structure of the flow emerges, plays a critical role in the architecture. In effect, our method reflects the fact that continuous-time systems are most common in practice, while the control signals typically arise from discrete-time computation.

A number of avenues for expanding our results are in sight. We have used one particular way of approximating the discrete-time system representing the flow function, namely RNNs. Other sequence models could be applied instead to create variations on our architecture. An interesting direction would be to impose stability conditions on the flow which enables the approximation to hold for unbounded times, as is done in [8] for continuous-time RNNs. Estimates on the number of parameters needed to achieve a given approximation quality would also be of interest. Finally, when training learning models in practice the mean squared loss (24) computed on a finite number of trajectory samples is minimised, which motivates an analysis of the sample complexity of learning the flow function.

X. Acknowledgments

This work was supported by the Swedish Research Council Distinguished Professor Grant 2017-01078 and a Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at C3SE partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

[1] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, “Neural ordinary differential equations,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Dec. 2018.

[2] E. De Brouwer, J. Simm, A. Arany, and Y. Moreau, “GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[3] G. I. Beintema, M. Schoukens, and R. T’oth, “Continuous-time identification of dynamic state-space models by deep subspace encoding,” in *International Conference on Learning Representations*, 2022.

[4] H. Garnier and P. C. Young, “Special issue on ‘Applications of Continuous-Time Model Identification and Estimation’,” *International Journal of Control*, vol. 87, no. 7, pp. 1317–1318, Jul. 2014.

[5] J. Schoukens and L. Ljung, “Nonlinear System Identification: A User-Oriented Road Map,” *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, Dec. 2019.

[6] E. D. Sontag, “Neural Nets as Systems Models and Controllers,” in *Seventh Yale Workshop on Adaptive and Learning Systems*, May 1992.

[7] X.-D. Li, J. Ho, and T. Chow, “Approximation of dynamical time-variant systems by continuous-time recurrent neural networks,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 10, pp. 656–660, Oct. 2005.

[8] J. Hanson and M. Raginsky, “Universal Simulation of Stable Dynamical Systems by Recurrent Neural Nets,” in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, Jul. 2020.

[9] J. Hanson, M. Raginsky, and E. Sontag, “Learning Recurrent Neural Net Models of Nonlinear Systems,” in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. PMLR, May 2021, pp. 425–435.

[10] A. Rahman, J. Drgoňa, A. Tuor, and J. Strube, “Neural Ordinary Differential Equations for Nonlinear System Identification,” in *2022 American Control Conference (ACC)*, Jun. 2022, pp. 3979–3984.

[11] M. Forgione and D. Piga, “Continuous-time system identification with neural networks: Model structures and fitting criteria,” *European Journal of Control*, vol. 59, pp. 69–81, May 2021.

[12] N. Geneva and N. Zabaras, “Transformers for modeling physical systems,” *Neural Networks*, vol. 146, pp. 272–289, Feb. 2022.

[13] D. Floryan and M. D. Graham, “Data-driven discovery of intrinsic dynamics,” *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1113–1120, Dec. 2022.

[14] S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016.

[15] P. Bevanda, S. Sosnowski, and S. Hirche, “Koopman operator dynamical models: Learning, analysis and control,” *Annual Reviews in Control*, vol. 52, pp. 197–212, Jan. 2021.

[16] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 2019.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, Jun. 2021.

[18] G. Kissas, J. H. Seidman, L. F. Guilhoto, V. M. Preciado, G. J. Pappas, and P. Perdikaris, “Learning Operators with Coupled Attention,” *Journal of Machine Learning Research*, vol. 23, no. 215, pp. 1–63, 2022.

[19] M. Biloš, J. Sommer, S. S. Rangapuram, T. Januschowski, and S. Günnemann, “Neural Flows: Efficient Alternative to Neural ODEs,” in *Advances in Neural Information Processing Systems*, Oct. 2021.

[20] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, “Fourier Neural Operator for Parametric Partial Differential Equations,” in *International Conference on Learning Representations*, 2021.

[21] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, “Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators,” *Nature Machine Intelligence*, vol. 3, no. 3, pp. 218–229, Mar. 2021.

[22] M. Aguiar, A. Das, and K. H. Johansson, “Learning Flow Functions from Data with Applications to Nonlinear Oscillators,” in *IFAC World Congress 2023*, Mar. 2023.

[23] E. D. Sontag, *Mathematical Control Theory*, ser. Texts in Applied Mathematics, J. E. Marsden, L. Sirovich, M. Golubitsky, and W. Jäger, Eds. New York, NY: Springer, 1998, vol. 6.

[24] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[25] A. M. Schäfer and H. G. Zimmermann, “Recurrent Neural Networks Are Universal Approximators,” in *Artificial Neural Networks – ICANN 2006*, ser. Lecture Notes in Computer Science, S. D. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Eds., 2006.

[26] D. Angeli and E. D. Sontag, “Forward completeness, unboundedness observability, and their Lyapunov characterizations,” *Systems & Control Letters*, vol. 38, no. 4, pp. 209–217, Dec. 1999.

[27] R. Sepulchre, G. Drion, and A. Franci, “Excitable behaviors,” no. arXiv:1704.04989, Apr. 2017.