Input-to-State Stable Neural Ordinary Differential Equations with Applications to Transient Modeling of Circuits

Alan Yang ASYANG2@ILLINOIS.EDU
Jie Xiong JIEX2@ILLINOIS.EDU
Maxim Raginsky MAXIM@ILLINOIS.EDU
Elyse Rosenbaum ELYSE@ILLINOIS.EDU
University of Illinois
Urbana, IL 61801

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

This paper proposes a class of neural ordinary differential equations parametrized by provably input-to-state stable continuous-time recurrent neural networks. The model dynamics are defined by construction to be input-to-state stable (ISS) with respect to an ISS-Lyapunov function that is learned jointly with the dynamics. We use the proposed method to learn cheap-to-simulate behavioral models for electronic circuits that can accurately reproduce the behavior of various digital and analog circuits when simulated by a commercial circuit simulator, even when interconnected with circuit components not encountered during training. We also demonstrate the feasibility of learning ISS-preserving perturbations to the dynamics for modeling degradation effects due to circuit aging.

Keywords: Input to State Stability, Safe learning, Neural ODE, Circuit simulation

1. Introduction

We consider the problem of learning input-to-state stable (ISS) continuous-time dynamics from observed input and output trajectories. Stability constraints provide an inductive bias that can help a dynamics learning algorithm choose one model over another and may improve the generalization accuracy of the dynamical system model when it is simulated in novel interconnection configurations. In some cases, imposing a stability constraint can guide the learning algorithm and improve performance.

This work was motivated by a need for fast-to-simulate behavioral models of electronic circuits. A system-on-a-chip (SoC), or larger microelectronic system, is composed of multiple functional blocks, often referred to as *IP blocks*, where IP denotes intellectual property. Prior to manufacturing, simulation is used to verify system performance. Each IP block may be represented by its transistor-level *netlist*, which specifies the interconnection of transistors that comprise its nonlinear dynamics. Transient simulation of the complete model of the non-linear system can be prohibitively slow, so it is preferable to replace the transistor-level description of each IP block with a cheaper-to-simulate behavioral model.

Transient circuit simulation presents a challenge for dynamics learning. At test time, an adaptive timestep solver is used to simulate the learned dynamics model, which is usually interconnected (in feedback) with external circuits and other IP blocks (Hajj, 2016). We refer to these external elements collectively as the *load*. In order to be useful, the model must be accurate when interconnected with

a variety of loads, and those loads are generally not known a priori. Due to uncertainty over the load, a model that performs well in training and validation may fail when embedded in a circuit simulation at test time.

In this work, we assume that the circuit of interest is well-modeled by a system of ODEs. A natural approach is to directly learn a parametric system of controlled ODEs

$$\dot{x} = f(x, u), \tag{1}$$
$$y = h(x), \tag{2}$$

which has state $x \in \mathbb{R}^n$, input $u \in \mathbb{R}^m$, and output $y \in \mathbb{R}^p$. u and y consist of node voltages and currents, and possibly their time derivatives.

In this work, we directly learn a *neural ODE* model of the form (1) – (2), which may be trained by either directly backpropagating through an ODE solver or implicitly differentiating through the solution to the ODEs using an adjoint method (Chen et al., 2018). Our models are trained using interpolated trajectories of $u(t)$ and $y(t)$ obtained from a circuit simulator; this approach is similar to prior works on neural ODE models of physical systems with continuous inputs (Kidger et al., 2020; Zhong et al., 2020).

We focus on the case where f takes the form of a continuous-time recurrent neural network (CTRNN) and h is an affine function of the state. Models of this form are universal approximators on finite time intervals (Funahashi and Nakamura, 1993) and, in some cases, infinite time intervals (Hanson and Raginsky, 2020). In addition, a CTRNN may be implemented as a generic circuit block using the Verilog-A behavioral modeling language and subsequently simulated by commercial circuit simulators (Chen et al., 2017).

In Section 2, we propose a CTRNN parametrization that is guaranteed to be ISS with respect to a Lur'e-Postnikov type (quadratic plus integral) ISS-Lyapunov function $V(x)$, which has parameters that are learned jointly with the CTRNN model parameters. ISS is a natural *a priori* assumption for many circuits; the state is guaranteed to be bounded given bounded inputs, and the state converges to a unique equilibrium if the input is set to zero.

There has been recent interest in learning neural ODEs jointly with a Lyapunov-like function $V(x)$. For autonomous systems, a sufficient condition for global asymptotic stability (GAS) is to ensure that V is strictly decreasing along any system trajectory, i.e.,

$$\dot{V}(x) < 0 \quad \forall x \neq 0. \tag{3}$$

Richards et al. (2018) encourage (3) via regularization, but do not guarantee that the dissipation inequality holds everywhere. Kolter and Manek (2019) and Massaroli et al. (2020) define the model dynamics as a function of $V(x)$ such that (3) holds for all x. Cranmer et al. (2020) and Zhong et al. (2020) considered the related problem of learning dynamics with Lagrangian and Hamiltonian structure, respectively.

Our approach is similar to that of Kolter and Manek (2019) in the sense that we guarantee that a dissipation inequality on V holds everywhere. However, we consider ISS, which can be seen as a generalization of GAS to systems with inputs. Also, we focus on CTRNN model structures as opposed to arbitrary dynamics models. We use a stability condition that generalizes the ISS condition derived by Ahn (2011), which is based on a quadratic ISS-Lyapunov function. In another related work, Cao et al. (2006) first learn an unconstrained circuit model and subsequently stabilize the model using nonlinear constrained optimization. In contrast, we build the stability constraint directly into the model parametrization.

Besides providing stability guarantees, we observed that our proposed model parametrization can accelerate training convergence. In this sense, it is related to prior works on regularization methods for accelerating neural ODE training. For example, Finlay et al. (2020) penalized the complexity of the model dynamics while Kelly et al. (2020) penalized the forward ODE solution time. Unlike those methods, our stability constraint does not introduce additional penalty terms, which can be difficult to tune.

We also show that our ISS parametrization is directly compatible with *aging-aware* circuit modeling. The dynamics of a circuit drift over time due to semiconductor degradation. Aged dynamics, estimated using physics-based approaches (Tu et al., 1993), can be used to verify lifetime specifications and identify aging-induced failures. Rosenbaum et al. (2020) directly learn an aging-aware circuit model by choosing the dynamics f in (1) and output map h in (2) to themselves be learned functions of a periodic *stress waveform* u_{stress}, which is assumed to have been applied to the circuit continuously for an operating time T_{op} on the order of years. Aging analysis can greatly benefit from fast-to-simulate surrogate models since separate aging simulations are needed to characterize different possible use condition profiles, each of which is specified by a pair $(u_{\text{stress}}, T_{\text{op}})$.

Section 2 presents our ISS-constrained model and describes how it can be used for transient circuit simulation, with and without aging effects. Section 3 evaluates the proposed methods on a variety of circuit modeling tasks.

2. Input-to-State Stable Continuous-Time Recurrent Neural Networks

2.1. Continuous-Time Recurrent Neural Networks

We consider controlled neural ODEs of the form

$$\dot{x} = -\frac{1}{\tau}x + W\sigma_\ell(Ax + Bu + \mu) + \nu, \tag{4}$$

$$y = Hx + b, \tag{5}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, and $y \in \mathbb{R}^p$ is the output. $\tau > 0$ is a positive scalar time constant, and $W, A^\top \in \mathbb{R}^{n\times\ell}$, $B \in \mathbb{R}^{\ell\times m}$, $\mu \in \mathbb{R}^\ell$, and $\nu \in \mathbb{R}^n$ are parameters. The element-wise function $\sigma_\ell : \mathbb{R}^\ell \to \mathbb{R}^\ell$ has the form $\sigma_\ell(w) = \big[\sigma(w_1), \ldots, \sigma(w_\ell)\big]^\top$, where σ is a strictly increasing, continuous, and subdifferentiable scalar-valued nonlinearity that satisfies $\sigma(0) = 0$ and the slope condition

$$0 \le \frac{\sigma(r) - \sigma(r')}{r - r'} \le 1, \quad \forall r, r' \in \mathbb{R}, r \ne r'. \tag{6}$$

Geometrically, (6) means that the graph of σ lies within a sector in the first and third quadrants, between the horizontal axis and the line with slope one. For example, the conditions on $\sigma(\cdot)$ are satisfied by the rectified linear unit $\text{ReLU}(\cdot) = \max\{0, \cdot\}$ and the hyperbolic tangent $\tanh(\cdot)$.

The dynamics (4) may be interpreted as a feedforward neural network with a single hidden layer of dimension ℓ and a stabilizing term $-x/\tau$, which is similar to "skip-connections" in residual networks (He et al., 2016). We also assume that $\ell \ge n$. Universal approximation results guarantee that a dynamical system with state dimension n can be approximated arbitrarily well by a CTRNN of the form (4) – (5), as long as ℓ is sufficiently large (Funahashi and Nakamura, 1993; Hanson and Raginsky, 2020).

2.2. Input-to-State Stability

The notion of input-to-state stability (ISS) was developed as a state-space approach to analyzing the stability of systems with inputs (Sontag, 2008). Suppose that (4) has an equilibrium point and, without loss of generality, that the equilibrium is at the origin.

Definition 1 *The system (1) is input-to-state stable (ISS) if there exist a class $\mathcal{KL}$ function [1] β and class $\mathcal{K}_\infty$ function γ such that*

$$\|x(t)\| \leq \beta(\|x_0\|, t) + \gamma(\|u\|_\infty) \tag{7}$$

for all $t \geq 0$, given any bounded input $u : [0, \infty) \to \mathbb{R}^m$ and initial condition $x(0) = x_0$.

The ISS property captures the idea that bounded inputs result in bounded state. Moreover, the effect of the initial condition on the trajectory (the transient response) should diminish to zero as $t \to \infty$, with rate bounded by the function β. A sufficient condition for ISS can be found by identifying an appropriate ISS-Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}_+$.

Theorem 2 *(Khalil, 2002) The system (1) is ISS if it admits an ISS-Lyapunov function, i.e., a smooth, positive definite, and radially unbounded function V for which there exist a positive definite function α and class $\mathcal{K}$ function g such that, for bounded inputs u,*

$$\dot{V}(x, u) := \nabla V(x)^\top f(x, u) \leq -\alpha(x) \quad \text{if } \|x\| \geq g(\|u\|). \tag{8}$$

The dissipation inequality (8) ensures that V, and therefore $\|x\|$, cannot grow too large relative to the magnitude of the input. Note that in the absence of inputs, $g(0) = 0$, and Theorem 2 reduces to a sufficient condition for global asymptotic stability. In that case, we refer to the associated function V simply as a Lyapunov function.

2.3. Lyapunov Diagonal Stability Condition

Forti and Tesi (1995) derived a sufficient condition for which the CTRNN (4) is GAS for constant input $u(t) \equiv u_0$.

Proposition 3 *(Forti and Tesi, 1995) If the matrix A is full rank, i.e. $\operatorname{rank}(A) = n$, and there exists a positive diagonal matrix $\Omega = \operatorname{diag}(\omega_1, \ldots, \omega_\ell)$ with $\omega_i > 0$ for each $i = 1, \ldots, \ell$ such that*

$$\Omega\Big(AW - \frac{1}{\tau}I\Big) + \Big(W^\top A^\top - \frac{1}{\tau}I\Big)\Omega \prec 0, \tag{9}$$

then (4) is GAS for constant input $u(t) \equiv u_0$.

If the conditions of Proposition 3 hold, we say that the matrix $AW - (1/\tau)I$ is *Lyapunov Diagonally Stable* (LDS). The rank condition on A is not restrictive, since the set of rank-deficient A has measure zero. The proof of Proposition 3 makes use of a Lyapunov function of the form

$$V(x) = x^\top P x + 2\sum_{i=1}^{\ell} \omega_i \int_0^{A_i x} (\sigma_\ell)_i(r)\, \mathrm{d}r, \tag{10}$$

1. The function $\alpha : [0, T) \to [0, \infty)$ belongs to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$. If $\alpha \in \mathcal{K}$ with $T = \infty$ and $\lim_{r\to\infty} \alpha(r) = \infty$, then $\alpha \in \mathcal{K}_\infty$. Finally, $\beta : [0, T) \times [0, \infty) \to [0, \infty)$ belongs to class $\mathcal{KL}$ if, for each fixed t, $\beta(r, t) \in \mathcal{K}$, and for each fixed r, $\beta(r, t)$ is decreasing with respect to t such that $\beta(r, t) \to 0$ as $t \to \infty$.

where $P \succ 0$ is a positive definite matrix, A_i denotes the i^{th} row of A in (4) for each i, and $\omega_i \geq 0$. If V of the form (10) can be used to prove 0-GAS for (4), then it it can also serve as an ISS-Lyapunov function.

Proposition 4 *If the conditions in Proposition 3 are satisfied, then* (4) *is also ISS.*

The direct extension of Proposition 3 to ISS is a consequence of the fact that the LDS condition guarantees that (4) is globally exponentially stable when $u \equiv 0$. In general, 0-GAS is a necessary, but not sufficient, condition for ISS.

2.4. An Input-to-State Stable Model Parametrization

In this subsection, we introduce our main result, a CTRNN parametrization that is provably ISS. Observe that as the matrix AW approaches the zero matrix, the matrix on the left hand side of (9) approaches $-\frac{2}{\tau}\Omega$, which is negative definite. Therefore, we may stabilize a given CTRNN by scaling AW. Here, we consider a parametrized matrix $A_\theta \in \mathbb{R}^{\ell \times n}$, and define A to be

$$A = \frac{1}{\rho(\tau, A_\theta, W, \Omega) + 1} A_\theta, \tag{11}$$

In the following, let $\lambda_{\max}(M)$ denote the largest eigenvalue of a symmetric matrix M.

Theorem 5 *For any $\delta > 0$, let*

$$\rho(\tau, A_\theta, W, \Omega) = \text{ReLU}\left(\frac{\tau}{2}\lambda_{\max}\left(\Omega^{1/2} A_\theta W \Omega^{-1/2} + \Omega^{-1/2} W^\top A_\theta^\top \Omega^{1/2}\right) - 1 + \delta\right), \tag{12}$$

Then, (4) *with A given by* (11) *is ISS.*

Using (12), the scalar ρ is made large enough to ensure that the LDS condition holds, if the condition does not already hold with $A = A_\theta$. The hyperparameter δ controls the minimum dissipation rate of V; smaller values of δ allow for longer transients. The model may be directly trained using a gradient descent method, since the stability constraint is built into the definition of A. Ω parametrizes V in (10), and may be either learned with the rest of the model parameters or fixed, e.g. to $\Omega = I$.

2.5. Training CTRNN Neural ODEs

At the start of a transient simulation, a circuit simulator sets the initial condition of its state variables to an equilibrium point given the initial input. For the model (1) – (2), the circuit simulator sets $x(0) = x_0$, where x_0 satisfies

$$0 = f(x_0, u(0)). \tag{13}$$

This is done using a numerical root-finding scheme, such as the Newton-Raphson method. The equilibrium condition (13) does not necessarily uniquely define x_0, and a user-specified initial condition may need to be provided. Fortunately, if f is given by a CTRNN (4) satisfying the LDS condition, then Proposition 3 guarantees the existence of a unique x_0 that satisfies (13) for any $u(0)$. During model training, we set the initial condition by numerically solving (13). In order to obtain the necessary derivatives for optimization, we implicitly differentiate through the root-finding operation (Bai et al., 2019).

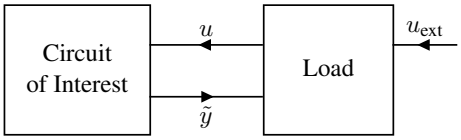


Figure 1: Simulation diagram for an interconnection between a circuit of interest and a load.

In our experiments, we consider the simulation setup illustrated in Figure 1. In a simulation, the circuit of interest is interconnected with a load, which itself may be driven by an external signal u_{ext}. We assume that both the load and u_{ext} are random and have known distributions. We train our models using N input and output trajectories $u^{(i)}$ and $\tilde{y}^{(i)}$ for $i = 1, \ldots, N$. Each pair $(u^{(i)}, \tilde{y}^{(i)})$ is obtained by simulating the system in Figure 1 on a time interval $[0, T]$ with initial condition defined by (13), using a random instantiation of the load and u_{ext}. Like Kidger et al. (2020), we obtain continuous trajectories $u^{(i)}$ and $\tilde{y}^{(i)}$ by interpolating the solution points provided by the circuit simulator.

The model parameter learning problem is given by the optimization problem

$$\text{minimize } \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\int_0^T \left(\tilde{y}^{(i)}(t) - y^{(i)}(t)\right)^2 \mathrm{d}t, \tag{14}$$

where $y^{(i)}$ is the predicted output. We estimate the integral in (14) by a Monte Carlo estimate as follows. Let S be a random variable uniformly distributed on $[0, T]$. Then, we have $\frac{1}{T}\int_0^T \left(\tilde{y}^{(i)}(t) - y^{(i)}(t)\right)^2 dt = \mathbb{E}[\left(\tilde{y}^{(i)}(S) - y^{(i)}(S)\right)^2]$ for each i, and so we may estimate the expectation using $\mathbb{E}[\left(\tilde{y}^{(i)}(S) - y^{(i)}(S)\right)^2] \approx \frac{1}{K}\sum_{j=1}^{K}\left(\tilde{y}^{(i)}(S_j) - y^{(i)}(S_j)\right)^2$, where $S_1, \ldots, S_K$ are i.i.d. copies of S.

2.6. Aging-Aware Neural ODEs

Conventionally, circuit aging simulation involves two transient simulations of the complete transistor-level netlist (Tu et al., 1993). In the first step, the fresh circuit ($T_{\text{op}} = 0$) is simulated subject to u_{stress} on a short time horizon $T_{\text{stress}} \ll T_{\text{op}}$[2] to estimate the per-transistor stress profile. Each transistor's dynamics is subsequently age-adjusted assuming that the T_{stress}-periodic input u_{stress} is applied for time T_{op}, which is typically on the order of years. In the second step, the circuit is re-simulated using the age-adjusted transistor models, subject to a possibly new input u.

Rosenbaum et al. (2020) proposed a two-step learning approach to learn aging-aware models; we extend that approach to include the ISS constraint. In the first step, we learn a "fresh" CTRNN

$$\begin{aligned}\dot{x} &= -\frac{1}{\tau_0}x + W_0\sigma_\ell(A_0x + B_0u + \mu_0) + \nu_0,\\ y &= H_0x + b_0,\end{aligned}$$

2. Modern circuits have nanosecond-scale signal periods; T_{stress} on the order of tens of nanoseconds is usually sufficient.

corresponding to $T_{\text{op}} = 0$ using the approach in Subsection 2.5. In the second step, we fix $\tau_0, W_0, \ldots, b_0$ and form an aging-aware CTRNN (4) – (5) whose parameters are given by

$$\begin{aligned} \tau &= \tau_0 + \Delta_\tau(u_{\text{stress}}, T_{\text{op}}), \\ W &= W_0 + \Delta_W(u_{\text{stress}}, T_{\text{op}}), \\ &\ldots \\ b &= b_0 + \Delta_b(u_{\text{stress}}, T_{\text{op}}), \end{aligned}$$

where $\Delta_\tau, \Delta_W, \ldots, \Delta_b$ are learned parameter perturbation functions. The model structure is suitable because, for realistic use conditions, the stress-induced drift in the dynamics will be relatively small, as illustrated by the example in Figure 2*(d)*. An ISS aging-aware model may be obtained by setting $A_\theta = A_0 + \Delta_A(u_{\text{stress}}, T_{\text{op}})$ in (11). The perturbation functions are learned using randomly-generated u_{stress}, T_{op}, and corresponding output trajectories $\tilde{y}$ obtained from the circuit simulator.

3. Experiments

3.1. Test Cases

We evaluated the proposed method using the following four test cases, which are similar to those used in prior works (Rajwardan, 2019; Chen, 2019; Rosenbaum et al., 2020), and the problem data were chosen to reflect realistic use conditions. The code used to train our models is available at https://github.com/syanga/ISSNeuralODE.

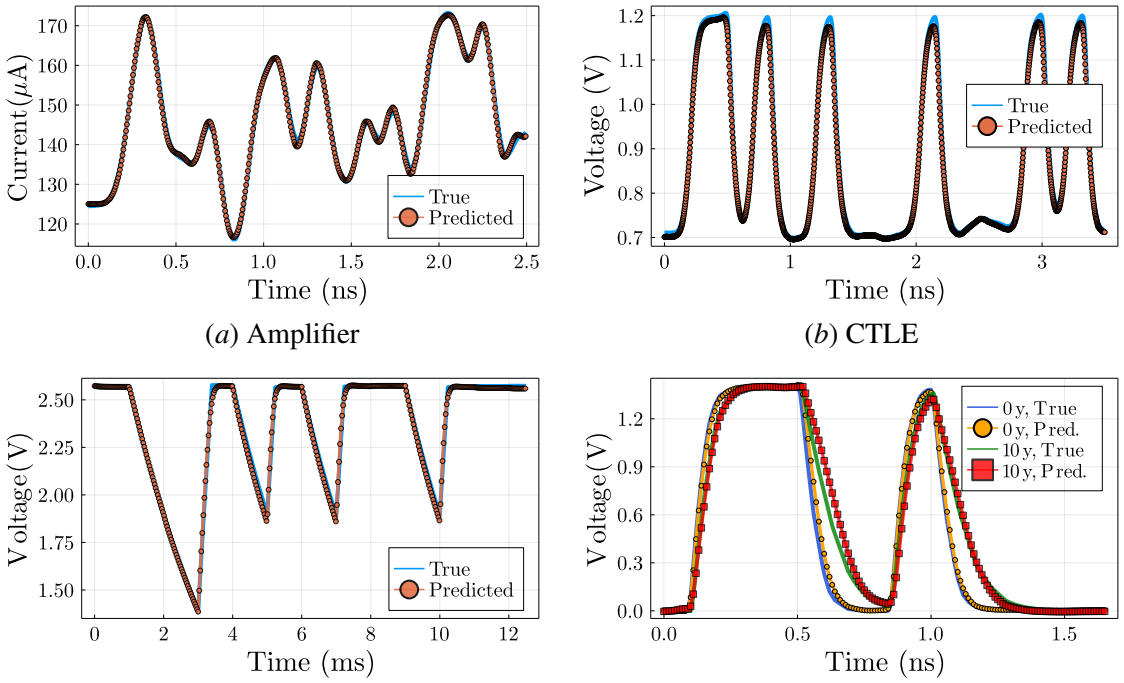
Common Source Amplifier. This circuit is a one-transistor, two-port voltage amplifier connected to resistor-capacitor (RC) loads with randomly generated values. The system input u_{ext} is driven by a random piecewise linear voltage source. The goal is to predict the currents at the input and output ports, given the port voltages. An example output current waveform is shown in Figure 2*(a)*. The learned CTRNN models had dimensions $n = 6$, $\ell = 14$, and $m = p = 2$.

Continuous-Time Linear Equalizer (CTLE). The CTLE is a five-transistor differential amplifier with two input ports and two output ports. It is designed to compensate for signal distortion that occurs when digital data are transmitted between two chips. The system input u_{ext} is given by the output of a pseudorandom bit sequence generator passed through a USB serial link, and the ports are connected to randomly-generated RC loads. The goal is to predict the input port currents and output port voltages, given the input port voltages, their time derivatives, and the output port currents. Figure 2*(b)* illustrates an example voltage waveform at one of the output ports. The learned CTRNN models had dimensions $n = 20$, $\ell = 30$, $m = 6$, and $p = 4$.

Large IP Block. This test circuit contains between 1000 and 2000 transistors; the exact number is unknown because the circuit is described by an encrypted netlist. Unlike the previous two test cases, this circuit is used with a known, fixed load. The goal is to predict two output voltages given seven input voltages. The inputs are continuous digital signals with fixed rise and fall times, and are driven by pseudorandom bit sequence generators. Figure 2*(c)* illustrates an example of one of the output responses. The learned CTRNN models had dimensions $n = 20$, $\ell = 30$, $m = 7$, and $p = 2$.

Inverter Chain with Aging Effects. The last test circuit is a chain of nine cascaded digital inverters; this circuit is often used to benchmark aging. For this test case, aging-induced degradation slows down the dynamics; Figure 2*(d)* illustrates the delay between the outputs of a fresh and aged

circuit. The model outputs two port currents given two port voltages, for a given stress profile $(u_{\text{stress}}, T_{\text{op}})$. The system input u_{ext} was driven by a random piece-wise linear voltage source, and random capacitive loads were connected to the output port. Aging analysis was performed using random piece-wise linear u_{stress} and random T_{op} sampled from a log uniform distribution from 0.001 to 10 years. We considered a special case of (4) with $W = I$ and $\nu = 0$ with dimensions $n = \ell = 20$ and $m = p = 2$, and learned parameter perturbations only for A, B, and μ. Δ_A, Δ_B, and Δ_μ were implemented by single-layer gated recurrent unit (GRU) network (Cho et al., 2014) with hidden state dimension 20.



(*a*) Amplifier (*b*) CTLE

(*c*) IP block (*d*) Inverter Chain

Figure 2: Examples of circuit responses simulated by a circuit simulator, using conventional circuit models (labeled as "True") and Verilog-A implementations of the learned CTRNNs satisfying the proposed ISS constraint (11) (labeled as "Predicted").

3.2. Results

We trained the CTRNNs by using the order three Bogacki-Shampine ODE solver with an interpolated adjoint method (Rackauckas et al., 2020) and the ADAM optimizer (Kingma and Ba, 2015). We took σ_ℓ to be ReLU with a bias term, and we used $\delta = 10^{-3}$ in the stability constraint (12). Each dimension of the inputs and outputs in the training data was separately normalized to $[-1, 1]$ prior to training, and the time horizon T was scaled up to be on the order of seconds (instead of nanoseconds). The models used in the first three test cases were trained using the Julia package `DiffEqFlux` (Rackauckas et al., 2020); the aging-aware models were trained using the Python

Table 1: MSE metrics. The reported MSE have been multiplied by a factor of 1000.

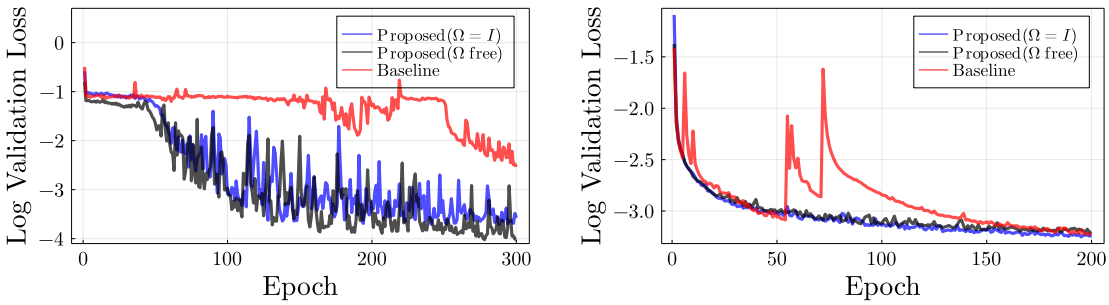
	Amplifier		CTLE		IP Block		Inverter Chain	
Model Type	Valid.	Test	Valid.	Test	Valid.	Test	Valid.	Test
Proposed (Ω free)	**0.234**	**0.263**	0.805	**0.939**	**0.031**	**0.250**	0.280	**0.918**
Proposed ($\Omega = I$)	0.260	0.314	1.03	4.44	0.142	0.2943	**0.232**	1.93
Baseline	0.243	0.279	**0.714**	3.1447	2.93	15.9	0.238	3.50

package `torchdiffeq` (Chen et al., 2018). Learned models were implemented in Verilog-A and simulated using the Spectre circuit simulator (Cadence Design Systems, Inc., 2020).

Table 1 compares three different training methods: CTRNN with no constraints (Baseline), the stability constraint (11) (Proposed, Ω free), and the stability constraint with Ω fixed to be the identity matrix (Proposed, $\Omega = I$). The table shows the mean squared error (MSE) of the predicted model outputs measured on a held-out validation set of input and output waveforms ("Valid.") and measured when the model is simulated by the circuit simulator as a Verilog-A model ("Test"). The "Test" MSE values are averaged across 100 simulations, each with random instantiations of load, u_{ext}, and stress profile ($u_{\text{stress}}, T_{\text{op}}$), in the aging-aware inverter chain test case. The MSE for both "Valid." and "Test" are computed after applying the aforementioned normalization to $[-1, 1]$. Figure 2 shows example simulations of the proposed ISS CTRNN models, carried out by Spectre.

Across all test cases, the errors on the held-out validation set were lower than the errors accumulated when tested in the circuit simulator. This is expected, since the model is simulated in open-loop with the validation set data rather than in closed-loop with the circuit simulator. At test time, the proposed method outperformed both the baseline learning method and the ISS model with Ω fixed to be the identity matrix. This demonstrates that our stability constraint can improve the generalization performance of the model in the circuit simulator.

For the IP block test case, the ISS constraint significantly accelerated the convergence of model training. This can be seen in the validation losses over the course of training in Figure 3*(a)*. Although the effect was less pronounced, Figure 3*(b)* shows how the ISS constraints were able to stabilize training for the CTLE test case as well.



(*a*) IP Block (*b*) CTLE

Figure 3: Comparison of validation losses over the course of training.

4. Conclusion

In this work, we introduced a neural ODE model parametrization that is provably ISS and can yield better models than baseline unconstrained models. In principle, our approach may be extended to impose other dissipation or invariance conditions based on, for example, passivity or energy conservation. It is also generically applicable to problem domains that benefit from constrained continuous-time dynamics learning, such as safety-critical robotics (Richards et al., 2018) and chemical process control Bonassi et al. (2020).

Appendix A. Omitted Proofs

A.1. Proof of Proposition 3

Consider any constant input $u(t) = u_0$. The change of coordinates $z = Ax - Bu_0 - \mu$ gives

$$\dot{z} = -\frac{1}{\tau}z + AW\sigma_\ell(z) + A\nu - \frac{1}{\tau}Bu_0 - \frac{1}{\tau}\mu. \tag{15}$$

Theorem 4 of Forti and Tesi (1995) guarantees that (15) has a unique equilibrium point z_0 which is GAS with respect to a Lyapunov function $V(z)$, where V is of the form (10). Since $\ell \geq n$ and A is full rank, $x_0 = (A^\top A)^{-1}A^\top z_0$ is the unique equilibrium point of (4), and is GAS with respect to $V((A^\top A)^{-1}A^\top z)$, which can also be written in the form (10).

A.2. Proof of Proposition 4

Suppose that (4) is 0-GAS with respect to the Lyapunov function (10). With $u \equiv 0$, the time derivative of V along trajectories of x is given by

$$\begin{aligned}\dot{V}(x) &= 2(x^\top P + \sigma_\ell(Ax)^\top \Omega A)(-x/\tau + W\sigma_\ell(Ax)) \\ &= \begin{bmatrix} x \\ \sigma_\ell(Ax) \end{bmatrix}^\top \begin{bmatrix} -2P/\tau & PW - \frac{1}{\tau}A^\top\Omega \\ W^\top P - \frac{1}{\tau}\Omega A & \Omega AW + W^\top A^\top \Omega \end{bmatrix} \begin{bmatrix} x \\ \sigma_\ell(Ax) \end{bmatrix}.\end{aligned} \tag{16}$$

Since the origin is a GAS equilibrium point, $\dot{V}(x) < 0$ for all $x \neq 0$, which implies that (16) is a negative definite quadratic form, i.e., there exists a $\lambda > 0$ such that $\dot{V}(x) \leq -\lambda(\|x\|^2 + \|\sigma_\ell(Ax)\|^2) \leq -\lambda\|x\|^2$. Due to the slope condition (6), $V(x)$ has a quadratic upper bound, and so by Theorem 4.10 of Khalil (2002), the unforced system is globally exponentially stable at the origin. Finally, Lemma 4.6 in Khalil (2002) gives ISS.

A.3. Proof of Theorem 5

To simplify the notation, we write $\rho(\tau, A_\theta, W, \Omega)$ as ρ with the arguments omitted. Since $\mathrm{ReLU}(\cdot) = \max\{0, \cdot\}$, we have $\rho \geq \frac{\tau}{2}\lambda_{\max}\big(\Omega^{\frac{1}{2}}A_\theta W\Omega^{-\frac{1}{2}} + \Omega^{-\frac{1}{2}}W^\top A_\theta^\top \Omega^{\frac{1}{2}}\big) - 1 + \delta$. Dividing both sides by $\rho + 1$ and rearranging gives $1 \geq \frac{\tau}{2}\lambda_{\max}\big(\Omega^{\frac{1}{2}}AW\Omega^{-\frac{1}{2}} + \Omega^{-\frac{1}{2}}W^\top A^\top\Omega^{\frac{1}{2}}\big) + \frac{\delta}{\rho+1}$, where $A = \frac{A_\theta}{\rho+1}$. Since $\lambda_{\max}(M)I \succeq M$ for symmetric M, $I \succeq \frac{\tau}{2}\big(\Omega^{\frac{1}{2}}AW\Omega^{-\frac{1}{2}} + \Omega^{-\frac{1}{2}}W^\top A^\top\Omega^{\frac{1}{2}}\big) + \frac{\delta}{\rho+1}I$. Finally, multiplying by $\Omega^{\frac{1}{2}}$ on the left and right sides and rearranging shows the LDS condition

$$\Omega\big(AW - \frac{1}{\tau}I\big) + \big(W^\top A^\top - \frac{1}{\tau}I\big)\Omega \preceq -\frac{2\delta}{\tau(\rho+1)}\Omega \prec 0.$$

Acknowledgments

This work was funded in part by the NSF under CNS 16-24811 and the industry members of the CAEML I/UCRC, and in part by the Illinois Institute for Data Science and Dynamical Systems (iDS^2), an NSF HDR TRIPODS institute, under award CCF-1934986.

References

Choon Ki Ahn. Robust stability of recurrent neural networks with ISS learning algorithm. *Nonlinear Dynamics*, 65(4):413–419, 2011.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, pages 690–701, 2019.

Fabio Bonassi, Enrico Terzi, Marcello Farina, and Riccardo Scattolini. LSTM neural networks: Input to state stability and probabilistic safety verification. In *Learning for Dynamics and Control*, pages 85–94, 2020.

Cadence Design Systems, Inc. Spectre circuit simulator reference. Technical report, 2020. URL http://web.engr.uky.edu/~elias/tutorials/Spectre/spectre_refManual.pdf.

Yi Cao, Runtao Ding, and Qi-Jun Zhang. State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis. *IEEE Transactions on Microwave Theory and Techniques*, 54(6):2398–2409, 2006.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.

Zaichen Chen. *Modeling of Electrical Circuit with Recurrent Neural Networks*. PhD thesis, University of Illinois at Urbana-Champaign, 2019.

Zaichen Chen, Maxim Raginsky, and Elyse Rosenbaum. Verilog-A compatible recurrent neural network model for transient circuit simulation. In *2017 IEEE 26th Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, pages 1–3, 2017.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M Oberman. How to train your neural ODE. *arXiv preprint arXiv:2002.02798*, 2020.

Mauro Forti and Alberto Tesi. New conditions for global stability of neural networks with application to linear and quadratic programming problems. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 42(7):354–366, July 1995.

Ken-Ichi Funahashi and Yuichi Nakamura. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, 6(6):801–806, 1993.

Ibrahim N. Hajj. *Computational Methods in Circuit Simulation*. CreateSpace Independent Publishing Platform, 2016.

Joshua Hanson and Maxim Raginsky. Universal simulation of stable dynamical systems by recurrent neural nets. In *Proceedings of Machine Learning Research*, volume 120, pages 384–392, 10–11 Jun 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Jacob Kelly, Jesse Bettencourt, Matthew James Johnson, and David Duvenaud. Learning differential equations that are easy to solve. *arXiv preprint arXiv:2007.04504*, 2020.

Hassan K Khalil. *Nonlinear Systems; 3rd ed.* Prentice-Hall, 2002.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *arXiv preprint arXiv:2005.08926*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference of Learning Representations (ICLR)*, 2015.

J. Zico Kolter and Gaurav Manek. Learning stable deep dynamics models. In *Advances in Neural Information Processing Systems 32*, pages 11128–11136. 2019.

Stefano Massaroli, Michael Poli, Michelangelo Bin, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Stable neural flows. *arXiv preprint arXiv:2003.08063*, 2020.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

Ashwarya Rajwardan. Receiver equalization for a 10 gigabit per second high-speed serial link in 65nm CMOS technology. Master's thesis, University of Illinois at Urbana-Champaign, 2019.

Spencer M Richards, Felix Berkenkamp, and Andreas Krause. The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. *arXiv preprint arXiv:1808.00924*, 2018.

E Rosenbaum, J Xiong, A Yang, Z Chen, and Maxim Raginsky. Machine learning for circuit aging simulation. In *2020 IEEE International Electron Devices Meeting (IEDM)*, pages 39–1. IEEE, 2020.

Eduardo D Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and Optimal Control Theory*, pages 163–220. Springer, 2008.

Robert H Tu, Elyse Rosenbaum, Wilson Y Chan, Chester C Li, Eric Minami, Khandker Quader, Ping K Ko, and Chenming Hu. Berkeley reliability tools-bert. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(10):1524–1534, 1993.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-net: Learning Hamiltonian dynamics with control. *International Conference on Learning Representations (ICLR)*, 2020.