CONTINUOUS-TIME IDENTIFICATION OF DYNAMIC STATE-SPACE MODELS BY DEEP SUBSPACE ENCODING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous-time (CT) modeling has proven to provide improved sample efficiency and interpretability in learning the dynamical behavior of physical systems compared to discrete-time (DT) models. However, even with numerous recent developments, the CT nonlinear state-space (NL-SS) model identification problem remains to be solved in full, considering common experimental aspects such as the presence of external inputs, measurement noise, latent states, and general robustness. This paper presents a novel estimation method that addresses all these aspects and that can obtain state-of-the-art results on multiple benchmarks with compact fully connected neural networks capturing the CT dynamics. The proposed estimation method, called the subspace encoder approach (SUBNET), ascertains these results by efficiently approximating the complete simulation loss by evaluating short simulations on subsections of the data, by using an encoder function to estimate the initial state for each subsection and a novel state-derivative normalization to ensure stability and good numerical conditioning of the training process. As part of our contributions, we prove that the use of subsections increases cost function smoothness, given the necessary requirements for the existence of the encoder function and we also show that the proposed state-derivative normalization is essential for reliable estimation of CT NL-SS models.

1 INTRODUCTION

Dynamical systems described by nonlinear state-space models with a state vector $x(t) \in \mathbb{R}^{n_x}$ are powerful tools of many modern sciences and engineering disciplines to understand potentially complex dynamical systems. One can distinguish between Discrete-Time (DT) $x_{n+1} = f(x_n, u_t)$ and Continuous-Time (CT) $\frac{dx(t)}{dt} = f(x(t), u(t))$ state-space models. In general, obtaining DT dynamical models from data is easier than CT models since data in computers is represented as discrete elements (e.g. arrays). However, the additional implementation complexity and computational costs associated with identifying CT models can be justified in many cases. First and foremost, from the natural sciences, we know that many systems are compactly described by CT dynamics which makes the continuity prior of CT models a well-motivated regularization/prior (De Brouwer et al., 2019). It has been observed that this regularization can be beneficial for sample efficiency (De Brouwer et al., 2019) which is a common observation when "including physics" in learning approaches (Karniadakis et al., 2021). Furthermore, the analysis of ODE equations is a well-regarded field of study with many powerful results and methods which could further improve model interpretability (Fan et al., 2021), such as applied in Bai et al. (2019). Another inherent advantage is that these models can be used with irregularly sampled or missing data (Rudy et al., 2019). Lastly, in the control community, CT models are generally regarded desirable for control synthesis tasks as shaping the behavior of the controller is much more intuitive in CT (Garcia et al., 1989). Hence, developing robust and general CT models and estimation methods would be greatly beneficial.

In the identification of physical CT systems, it is common to encounter challenges such as: external inputs ($u(t)$), noisy measurements, latent states, unknown measurement function/distribution (e.g. $y(t) = h(x(t))$), the need for accurate long-term predictions and a need for a sufficiently low computational cost. For instance, all these aspects need to be considered for the cascade tank benchmark problem (Schoukens & Noël, 2017). These aspects and the considered CT state-space model is summarized in Figure 1. Many of these aspects have been studied independently, for instance, Brajard et al. (2020); Rudy et al. (2019) explicitly addressed the presence of noise on the measurement

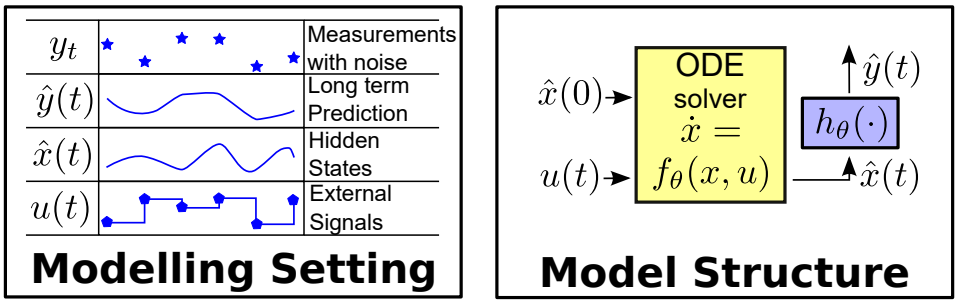

Figure 1: In this work, we consider the problem of estimating continuous-time (CT) state-space models from noisy observation (additive noise) with long-term prediction capabilities, hidden states and external signals in a computationally efficient and robust manner.

data, Maulik et al. (2020); Chen et al. (2018) provided methods for modeling dynamics with latent states, Zhong et al. (2020) considers the presence of known external inputs, Zhou et al. (2021a) provides a computationally tractable method for accurate long-term sequence modeling. However, formulating models and estimation methods for the combination of multiple or all aspects is in comparison underdeveloped with only a few attempts such as Forgione & Piga (2021a) that have been made.

In contrast to previous work, we present a CT encoder-based method which is a general, robust and well-performing estimation method for CT state-space model identification. That is, the formulation addresses noise assumptions, external inputs, latent states, an unknown output function, and provides state-of-the-art results on multiple benchmarks of real systems. The presented subspace encoder method is summarized in Figure 2. The proposed method considers a cost function evaluations on only short subsections of the available dataset which reduces the computational complexity. Furthermore, we show theoretically that considering subsections enhances cost function smoothness and thus optimization stability. The initial states of these subsections are estimated using the encoder function for which we present necessary requirements for its existence. Lastly, we introduce a normalization of the state and state-derivative and we show that it is required for proper CT estimation. Moreover, we attain additional novelty as these results are obtained without needing to impose a specific structure on the state-space (such as in Greydanus et al. (2019); Cranmer et al. (2020)) obtaining a practically widely applicable method.

Our main contributions are the following;

- We formally derive the problem of CT state-space model estimation with latent states, external inputs, and measurement noise.
- We reduce the computational loads by proposing a subspace encoder-based identification algorithm that employs short subsections, an encoder function that estimates the initial latent states of these subsections, and a state-derivative normalization term for robustness.
- We make multiple theoretical contributions; *(i)* we prove that the use of short subsections increases cost function smoothness by Lipschitz continuity analysis, *(ii)* we derive necessary conditions for the encoder function to exist and *(iii)* we show that a state-derivative normalization term is required for proper CT model estimation.
- We demonstrate that the proposed estimation method obtains state-of-the-art results on multiple benchmarks.

2 Related work

One of the most influential papers in CT model estimation is the introduction of neural ODEs (Chen et al., 2018), which showed that residual networks is presented as an Euler discretization of a continuous in-depth neural network. Moreover, they also show that one can employ numerical integrators to integrate through the depth in a computationally efficient manner. This depth can be interpreted as the time direction to be able to model dynamical systems. The ideas in the neural ODE contribution have been extended to/used in, for instance, normalizing flows to efficiently model arbitrary

probability distributions (Papamakarios et al., 2021; Grathwohl et al., 2019), and enhance the understanding and interpretability of neural networks (Fan et al., 2021). However, the neural ODE does not scale well for long sequences, nor does it consider external inputs or noise, and the optimization process is often unstable.

An adjacent research direction is the method/models which consider CT dynamics and directly use the state derivatives and even often the noiseless states to formulate structured and interpretable models such as Hamiltonian Neural Networks (HNN) (Greydanus et al., 2019), Lagrangian Neural Networks (LNN) (Cranmer et al., 2020) and Sparse Identification of Nonlinear Dynamics (SINDy) (Brunton et al., 2016). In contrast, the proposed method is formulated for an unstructured state-space and does not require the system state or the state derivatives to be known.

Our method is in part related to (Ayed et al., 2019) which concerns the estimation of CT models with latent variables. They also employ an encoder function to estimate initial states, however, this encoder is only dependent on the past outputs, contains a partially known state and there is no theoretical support for the method. Furthermore, in that work, only a fixed output function is considered and the involved optimization problem is solved as an optimal control problem whereas our formulation alters the simulation loss function to obtain a computationally desirable form. Furthermore, (Forgione & Piga, 2021a), to which we compare in this work, considers CT model with latent variables, subsections, and an additional loss term for the integration error. However, they include the initial states of these subsections as free optimization parameters. This increases the model complexity with the number of subsections. In contrast, our proposed method uses an encoder to estimate the initial states. This results in fixed model complexity. Furthermore, we only employ a single loss function and a novel state-derivative normalization term. Additionally, we provide theoretical insights into these existing elements and extend them to the considered setting in a robust and computationally efficient manner.

3 PROBLEM STATEMENT

Consider a system represented by a continuous-time nonlinear state-space (CT NL-SS) description sampled at a fixed interval Δt for simplicity:

$$\begin{aligned} \dot{x}_s(t) &= f(x_s(t), u(t)), \\ y_k &= h(x_{s,k}, u_k) + w_k, \end{aligned} \tag{1}$$

where the subscript notation denotes sampling as $x_{s,k} = x_s(k\Delta t)$, $x_s(t) \in \mathbb{R}^{n_{x_s}}$ is the system state variable, $u(t) \in \mathbb{R}^{n_u}$ is the input, $y_k \in \mathbb{R}^{n_y}$ is the output, f represents the system dynamics and h gives the output function while $w_k \in \mathbb{R}^{n_y}$ is a i.i.d. zero-mean white noise process with finite variance Σ_w.

For this system, the CT model estimation problem can be expressed for a given dataset of measurements:

$$D_N = \{(u_0, y_0), (u_1, y_1), ..., (u_{N-1}, y_{N-1})\},$$

with unknown w_k, $x_s(t)$, $\dot{x}_s(t)$, $\dot{y}_k$ and initial state $x_s(0)$, as the following optimization problem (a.k.a. simulation loss minimization):

$$\begin{aligned} \min_{\theta, x(0)} \quad & \frac{1}{N} \sum_{k=0}^{N-1} \|y(k\Delta t) - \hat{y}(k\Delta t)\|_2^2, \\ \text{s.t.} \quad & \hat{y}(t) = h_\theta(x(t)), \\ & \dot{x}(t) = f_\theta(x(t), u(t)), \end{aligned} \tag{2}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the model state, h_θ and f_θ are the output and state-derivative functions parameterized by θ and being Lipschitz continuous in their inputs and parameterization. These two functions are formulated as multi-layer feedforward neural networks during our experiments.

To obtain the simulation output $\hat{y}(k\Delta t)$, one can integrate $\dot{x}(t) = f_\theta(x(t), u(t))$ starting from the initial state $x(0)$. This integration can be performed with any ODE solver that allows for back-propagation such as Euler ($x(t + \Delta t) = x(t) + \Delta t f_\theta(x(t), u(t))$), RK4, or numerous adaptive step methods (Chen et al., 2018; Ribeiro et al., 2020).[1] To make this a well-posed optimization problem,

[1]The adjoint methods for gradient computation is not within the scope of this research.

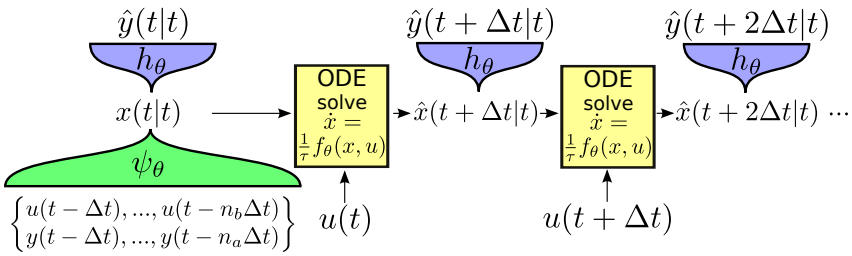

Figure 2: The CT subspace encoder (SUBNET) method applied on a subsection of the data of length $T\Delta t$ starting from time t. The encoder ψ_θ estimates the initial state, h_θ provides the output predictions while $\frac{1}{\tau} f_\theta$ governs the state dynamics. All three functions are parameterized by fully connected neural networks. Lastly, the $\frac{1}{\tau}$ factor is the novel state-derivative normalization factor which significantly increases optimization stability. All three functions are optimized together by minimizing the mean squared difference of the model outputs of multiple subsections of the available training data as seen in Eq. (3) which both reduces the computational cost and enhances cost function smoothness as seen in Theorem 1.

additional information or an assumption on the inter-sample behavior of $u(t)$ is required, since, for example, $u(\Delta t/2)$ is not present in D_N. This behavior is often chosen to be Zero-Order Hold (ZOH) (Ljung, 1999) as can be viewed in Figure 1.

Multiple major issues are encountered when solving the optimization Problem (2) with a gradient-descent-based method. The first issue is that computing the value of the loss function requires a forward pass on the whole length of the dataset (Ayed et al., 2019). Hence, the computational complexity grows linearly with the length of the dataset. Furthermore, a common occurrence is that the values of $x(t)$ or its gradient grows exponentially which results in non-smooth loss functions or gradients. This causes gradient-based optimization algorithms to become unreliable since the optimization process might be unstable or it converges to a local minima (Ribeiro et al., 2020). All these issues are addressed in the proposed method.

4 Proposed method

We propose to consider multiple overlapping short subsections of length $T\Delta t$ to form a truncated simulation loss instead of simulating over the entire length of the dataset. We express this in the following optimization problem (note that we express the optimization problem with discrete-time notation ($u_k := u(k\Delta t)$) for brevity):

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & \frac{1}{N-T-\max(n_a,n_b)+1} \sum_{n=\max(n_a,n_b)}^{N-T} \frac{1}{T} \sum_{k=0}^{T-1} \|y_{n+k} - \hat{y}_{n+k|n}\|_2^2, \\ \text{s.t.} \quad & \hat{y}_{n+k|n} = h_\theta(x_{n+k|n}), \\ & x_{n+k+1|n} = \text{ODEsolve}[\frac{1}{\tau} f_\theta, x_{n+k|n}, u_{n+k}, \Delta t], \\ & x_{n|n} = \psi_\theta(u_{n-1}, ..., u_{n-n_b}, y_{n-1}, ..., y_{n-n_a}). \end{aligned} \tag{3}$$

Here, the pipe (|) notation indicates the current index and the starting index as (current index | start index) to differentiate between different subsections. This pipe notation is similar to the notation used in Kalman filtering and conditional probability distributions (Chui et al., 2017). Furthermore, ODEsolve indicates a numerical scheme which integrates $1/\tau\ f_\theta(x, u)$ from the initial state $x_{n+k|n}$ for a length of Δt given the input u_{n+k}. Lastly, we introduced an encoder function ψ_θ with encoder lengths n_a and n_b, for the past output and input samples respectively, which estimates the initial states of the considered subsection. This encoder function will also be parameterized as a feedforward neural network during our experiments. A graphical summary of the proposed method called the CT subspace encoder approach (abbreviated as SUBNET) can be viewed in Figure 2.

The first observation is that optimization Problem (3) is a generalisation of (2) since if $T = N$ and $n_a = n_b = 0$, the original optimization Problem (2) is recovered. However, as one might observe,

this optimization problem is less computationally challenging to solve if $T < N$ since the first sum can be computed in parallel. In other words, computational costs scale as $\mathcal{O}(T)$ for (3) and $\mathcal{O}(N)$ for (2). Moreover, the smoothness of the encoder cost function is also enhanced since the associated Lipschitz constant $L_{V^{(\text{enc})}}$ can scale exponentially with the subsection length ($T\Delta t$) as shown in Theorem 1. The enhanced smoothness is reflected in the ease of optimization (Ribeiro et al., 2020).

Theorem 1. *The Lipschitz constant $L_{V^{(enc)}}$ of the cost function* (3)

$$\|V^{(enc)}(\theta_1) - V^{(enc)}(\theta_2)\|_2 \leq L_{V^{(enc)}}\|\theta_1 - \theta_2\|_2 \tag{4}$$

scales as

$$L_{V^{(enc)}} = \mathcal{O}(\exp(2T\Delta t L_f/\tau)) \tag{5}$$

where L_f is the Lipschitz constant of f_θ.

Proof. See Appendix 8.1 □

An error in the initial state of $x_{n|n}$ can significantly bias the estimate due to the short nature of the subsections. To counter this, we formulated an encoder function ψ_θ which estimates the initial state of each subsection. We do not add any additional loss term since an improved initial state error estimate also minimizes the transient error (Forgione & Piga, 2021a) which is present in the encoder loss.

A natural question to ask is under which conditions there exists an encoder function that can map from the past inputs and outputs to this initial state. In Appendix 8.2, we formally derive necessary conditions for the existence. These necessary conditions are that, state derivative f_θ requires to be Lipschitz continuous in x, and if the number of considered past outputs n_a and inputs n_b are equal then $n_a \geq n_x/n_y$ needs to be satisfied, among other conditions.

It is widely known that input and output normalization is essential for obtaining competitive models throughout deep learning in terms of respecting the prior assumptions made in for instance Xavier Weight Initialization (Glorot & Bengio, 2010). Input and output normalization can be seen to be insufficient when considering CT state-space model due to the presents of the hidden state $x(t)$ and the state-derivative $f(x, u)$. However, as shown in Theorem 2, any CT system can be transformed to become normalized by the introduction of a state transform and a positive $1/\tau$ normalization factor.

Theorem 2. *For any system of the form $\dot{x}(t) = f(x(t), u(t))$ with finite $x(t)$ and $\dot{x}(t)$ there exists a τ and a state transformation $\gamma\tilde{x}(t) = x(t)$ such that both the state $\tilde{x}$ and state-derivative $\tilde{f}$ of the transformed system $\dot{\tilde{x}} = \frac{1}{\tau}\tilde{f}(\tilde{x}(t), u(t))$ are normalized as in*

$$\textit{RMS}(\tilde{x}) = \sqrt{\frac{1}{L}\int_0^L \frac{1}{n_x}\|\tilde{x}(t)\|_2^2 dt} = 1 \qquad \& \qquad \textit{RMS}(\tilde{f}) = 1. \tag{6}$$

Proof. With

$$\gamma = \text{RMS}(x) \qquad \& \qquad \frac{1}{\tau} = \frac{\text{RMS}(\dot{x})}{\text{RMS}(x)} \tag{7}$$

the normalization conditions are satisfied, as shown below

$$\text{RMS}(\tilde{x}) = \text{RMS}(x)/\gamma = 1 \tag{8a}$$

$$\text{RMS}(\tilde{f}) = \tau\text{RMS}(\dot{\tilde{x}}) = \tau\text{RMS}(\dot{x})/\gamma = \tau\text{RMS}(\dot{x})/\text{RMS}(x) = 1 \tag{8b}$$

□

Hence, to assure the existence of a properly normalized model where both the state-derivative $\tilde{f}$ function and state $\tilde{x}$ are normalized, it is sufficient to include a state and state-derivative normalization factor. Furthermore, this proof also guides the choice of τ since the amplitude of $\text{RMS}(x)/\text{RMS}(\dot{x})$ might be known from physical insight or by an approximate model.

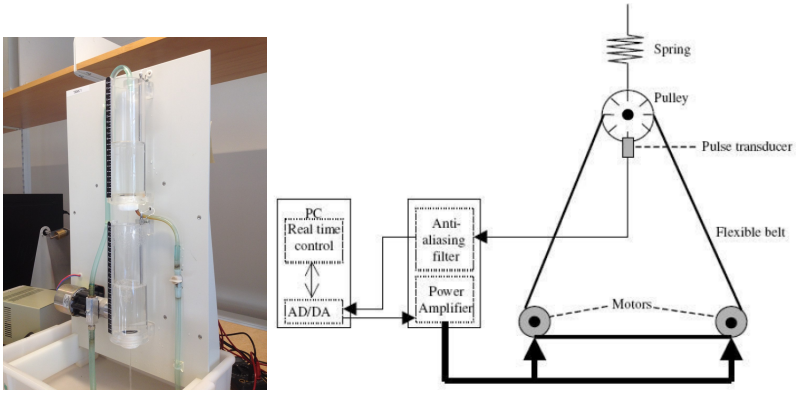

Figure 3: A photo of the **Cascade Tank with overflow (CCT)** system (Schoukens & Noël, 2017) and a graphical depiction of the **Coupled Electric Drive (CED)** system (Wigren & Schoukens, 2017). These two systems and the EMPS system are the basis of the benchmarks used in the analysis and comparison of the SUBNET method.

5 EXPERIMENTS

5.1 BENCHMARK DESCRIPTIONS

The **Cascade Tank with overflow (CCT)** benchmark (Schoukens & Noël, 2017; Schoukens et al., 2017) consists of measurements taken from a two-tank fluid system with a pump. The input signal controls a water pump that delivers water from the reservoir to the upper tank. Through a small opening in the upper tank, the water enters the lower tank where the water level is recorded. Lastly, through a small opening in the lower tank, the water re-enters the reservoir. This benchmark is nonlinear as the flow rates are governed by square root relations and the water can overflow either tank which is a hard saturation type of nonlinearity. The benchmark consists of two datasets with measurements of 1024 samples each at a sample rate of $\Delta t = 4\text{s}$. The first dataset is used for training, the first 512 samples of the second set are used for validation (used only for early stopping) and the entire second set for testing. Most of the other methods to which we compare our approach use the entire second set as validation and test set, as no explicit test set is provided in this benchmark description.

The **Coupled Electric Drive (CED)** benchmark (Wigren & Schoukens, 2017) consists of measurements from a belt and pulley with two motors where both clockwise and counter-clockwise movement is permitted. The motors are actuated by the given inputs and the measured output is a pulse transducer that only measures the absolute velocity (i.e. insensitive to the sign of the velocity) of the belt. The system approximately has three states; the velocity of the belt, the position of the pulley, and the velocity of the pulley. The benchmark consists of two datasets of measured 500 samples each at a sample rate of $\Delta t = 20\text{ms}$. The first 300 samples are used for training and the other 200 samples are for testing, of those samples the first 100 samples are also used for validation with both datasets. Similar to the last benchmark, even with this overlap, it is still a fair comparison as most of the other methods to which we compare use the entire second set as validation and test.

The **Electro-Mechanical Positioning System (EMPS)** benchmark (Janot et al., 2019) consists of measured signals from a one-dimensional drive system used to drive the prismatic joint of robots or machine tools. The provided measurements of the position are obtained in closed-loop actuated and no direct velocity measurements are available. The main source of nonlinearity are the nonlinear friction effects (e.g. static and dynamic friction). The benchmark consists of two sequences of samples 24841 with a sampling time of $\Delta t = 1\text{ms}$. As prescribed by the benchmark, the first sequence is used for training and validation and the second sequence for testing (i.e. the validation set and test set are completely disjoint). Specifically for the CT subnet implementation, we utilize 17885 samples of the first set for training and the last 6956 samples are used for validation while the entire second set is used for testing.

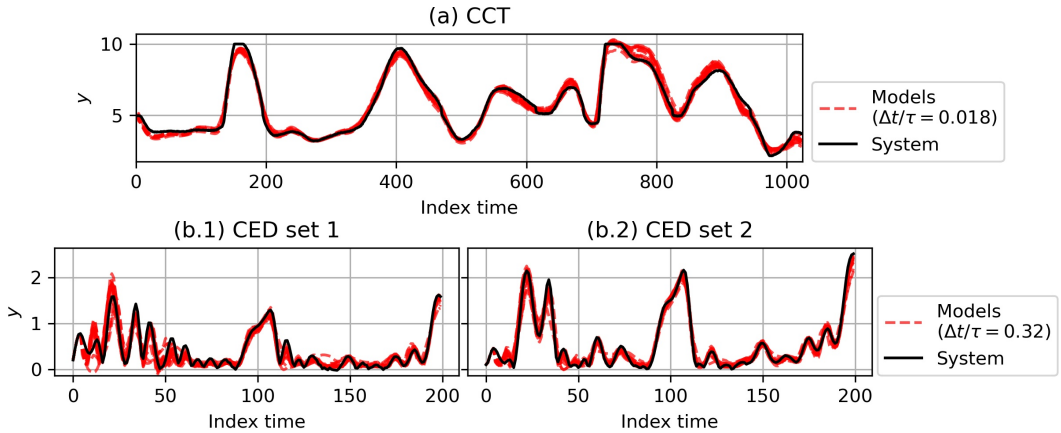

Figure 4: Time-domain simulation for both the (a) CCT and (b) CED benchmarks of the obtained models by the CT SUBNET method along the test set. Since the CED benchmark contains two separate test sequences, hence, they are shown in two separate figures.

5.2 Results

Using the SUBNET method, we estimate models where the three functions h_θ, f_θ and ψ_θ are implemented as 2 hidden layer neural networks with 64 hidden nodes per layer, tanh activation and a linear bypass from the input to the output for CCT and CED and 1 hidden layer with 30 hidden nodes for EMPS. As an ODE solver, we use a single RK4 step between samples and assume that the input signal has been applied in a zero-order hold sense. As for the implementation of the CT subspace encoder-based method, the following hyper-parameters are considered; $n_x = 2$, $n_a = n_b = 5$ and $T = 30$ for CCT, $n_x = 3$, $n_a = n_b = 4$ and $T = 60$ for CED and $n_x = 3$, $n_a = n_b = 20$ and $T = 200$ for EMPS. These hyperparameters are chosen based on the analysis methods shown in (Beintema et al., 2021) for discrete-time which has been validated to still hold for our case. The training is done by using the Adam optimizer with default settings (Kingma & Ba, 2015) with a batch size of 32 for CED, 64 for CCT and 1024 for EMPS and using a simulation on the validation dataset for early stopping to reduce overfitting.

We also directly compare our method with a reproduction of neural ODE on both benchmarks. We adapt the code and the example ("`latent_ODE.py`") available online (Chen et al., 2018) to include ZOH inputs, leaving the neural network unaltered and an RK4 integrator. We observed that the initial model was unstable for CCT and underperforming for CED and, hence, the neural ODE method alone was unable to provide state-of-the-art results. To stabilize and improve the neural ODE method we also introduce a state-derivative normalization term $1/\tau$ motivated by Theorem 2. The value of $1/\tau$ for CCT and CED for neural ODE was initially chosen to be the optimal value found in the SUBNET approach, however, in the CED case, it was lowered due to optimization instabilities. [2]

We compared our obtained model to the literature in Table 1 for CCT and CED. We report both the mean and the minimum of an ensemble of models estimated only differing in parameter initialization. This ensemble consists of 17 SUBNET models for both CCT and CED and 24 and 8 neuralODE models for CCT and CED respectively. The table also includes the discrete-time (DT) subspace encoder which has the same network structure and loss function as the CT subspace encoder but where the ODE solver is replaced by f_θ. The table shows that the obtained models with the CT subspace encoder method provide state-of-the-art results. The obtained models are the best-known with a black-box modeling approach on both benchmarks. Furthermore, we use unrestricted state-space and fully connected neural networks as model elements. Remarkably, the resulting performance is close to the performance of a grey-box model. Furthermore, Figure 4 illustrates that the resulting models have been able to model the nonlinear behavior present in both benchmarks.

[2]The code used for both SUBNET and neural ODE experiments is available at *"Github link removed for double-blind reviews, see zip file from supplementary materials until de-anonymization"*

Table 1: The test RMSE simulation on two benchmarks for the CT SUBNET method using an ensemble of models. The value given is the best RMSE simulation of all estimated models and the value between parentheses is the mean performance of the models. Note that we are unable to report the results for the neural ODE without state-derivative normalization $1/\tau$ for CCT since the optimization was unstable.

(a) CCT benchmark

Method	**RMSE**
BLA (Relan et al., 2017)	0.75
Volterra model (Birpoutsoukis et al., 2018)	0.54
State-space with GP-inspired prior (Svensson & Schön, 2017)	0.45
SCI (Forgione & Piga, 2021a)	0.40
NL-SS + NLSS2 (Relan et al., 2017)	0.34
TSEM (Forgione & Piga, 2021a)	0.33
Tensor B-splines (Karagoz & Batselier, 2020)	0.30
Vanilla neural ODE (Chen et al., 2018)	NaN
neural ODE with normalization ($\Delta t/\tau = 0.03$)	0.18 (0.33)
Grey-Box with physical overflow model (Rogers et al., 2017)	0.18
DT subspace encoder	0.37 (0.97)
CT subspace encoder ($\Delta t/\tau = 0.032$)	**0.22 (0.30)**

(b) CED benchmark

Method	**RMSE [ticks/s]**	
	Set 1	**Set 2**
RBFNN - FSDE (Ayala et al., 2014)	0.130	0.185
GP with rational quadratic kernel (Zhou et al., 2021b)	0.150	0.167
GP with squared exponential kernel (Zhou et al., 2021b)	0.153	0.132
Sparse Bayesian MLP (Zhou et al., 2021b)	0.149 (0.187)	0.120 (0.134)
Cascaded Splines (Scarpiniti et al., 2015)	0.216	0.110
Sparse Bayesian LSTM (Zhou et al., 2021b)	0.121 (0.155)	0.097 (0.126)
Extended Fuzzy Logic (Sabahi & Akbarzadeh-T, 2015)	0.150	0.092
Vanilla neural ODE (Chen et al., 2018)	0.141 (0.362)	0.098 (0.370)
neural ODE with normalization ($\Delta t/\tau = 0.12$)	0.131 (0.198)	0.086 (0.158)
DT subspace encoder	0.169 (0.187)	0.117 (0.1557)
CT subspace encoder ($\Delta t/\tau = 0.3$)	**0.115 (0.143)**	**0.074 (0.100)**

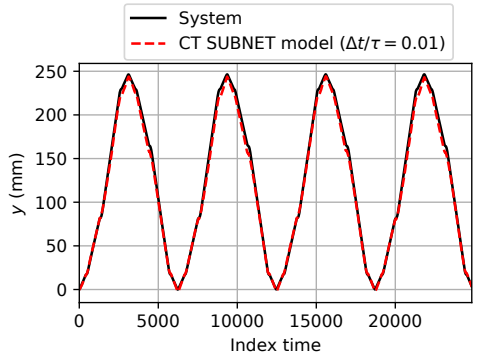

Figure 5: Time domain simulation results on the EMPS benchmark test set using the model obtained from the CT SUBNET method.

Table 2: The resulting test RMSE simulation on the test set for the EMPS benchmark compared with results in the literature.

Method	**RMSE [mm]**	**Fully Black-box**
BLA (Janot et al., 2019)	61.67	✓
dynoNET (Forgione & Piga, 2021b)	2.64	✗
CT SI ANN (Forgione & Piga, 2021a)	8.26	✓
IO stable CT ANN (Weigand et al., 2021)	3.68	✓
TSEM (Forgione & Piga, 2021a)	4.73	✗
SCI (Forgione & Piga, 2021a)	6.65	✗
Vanilla neural ODE (Chen et al., 2018)	NaN	✓
CT subspace encoder ($\Delta t/\tau = 0.01$)	4.61	✓

Table 1 also contains the results of the modified neural ODE with normalization. One observation is that the best and mean performance difference is significantly larger than for SUBNET. We think that this is due to the availability of only a single sequence in the training set for the CCT benchmark (and two sequences for CED) which results in extensive overfitting. In comparison, the subspace encoder method is less prone to overfitting since it uses many subsections of the available sequence(s). Moreover, the subspace encoder method only requires about 20 minutes to train a model to the lowest validation loss, whereas, neural ODE requires about 2 hours for CED and 5 hours for CCT.

There are considerably fewer black-box methods to compare for the EMPS benchmark. Nevertheless, we compare the CT SUBNET method to the existing methods in Table 2 and show the simulated response of the obtained model on the test set in Figure 5. The obtained results show a remarkable accuracy over the entire 24841 samples showing that the CT SUBNET method is indeed able to make accurate long-term predictions even while using relatively short sub-sequences of only $T = 200$ in length during training. Furthermore, since the validation and test set are disjoint we also show that overfitting does not play a role in the reported results.

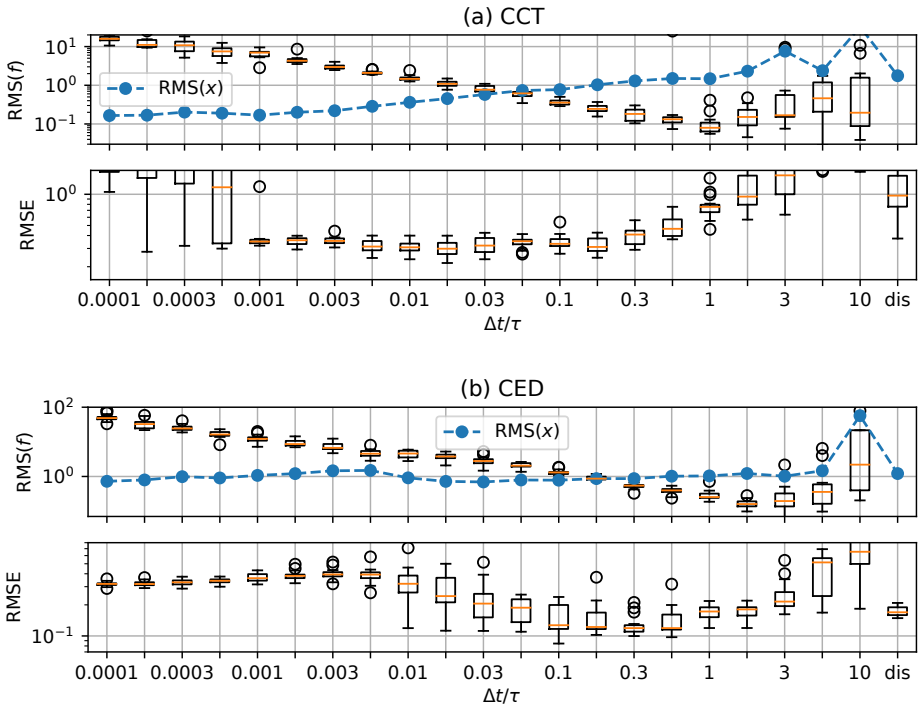

Figure 6: The influence of the state-derivative normalization hyperparameter $\Delta t/\tau$ as in $\dot{x} = \frac{1}{\tau} f_\theta(x, u)$ on different model properties for both the (a) CCT and (b) CED benchmarks. The shown model properties are the mean state amplitude $\text{RMS}(x)$, mean state-derivative amplitude $\text{RMS}(f)$ and the RMSE simulation on the test set(s). These two figures show that there exists a range of $\Delta t/\tau$ where $\text{RMS}(f) \approx \text{RMS}(x) \approx 1$ which numerically validates Theorem 2. Furthermore, $\Delta t/\tau$ with this property also has a significantly lowered RMSE simulation as was argued in it introduction of this hyperparameter.

When we introduced the state-derivative normalization factor $1/\tau$, we argued that it would normalize f_θ and that it would increase optimization stability and, hence, the quality of the obtained models. Here, we provide some empirical insight for these two statements by providing a parameter sweep over $\Delta t/\tau$. To eliminate variations due to different initial parameters we trained an ensemble of models which creates box plots with mean state amplitude defined by $\text{RMS}(x) \triangleq \sqrt{\frac{1}{N n_x} \sum_k \|x(k\Delta t)\|_2^2}$, mean state-derivative amplitude $\text{RMS}(f)$, and the RMSE simulation. These box-plots as shown in Figure 6 indeed illustrate that there exists an $1/\tau$ such that both $\text{RMS}(f) \approx \text{RMS}(x) \approx 1$ and that the best performing models are close to that value of $1/\tau$. Moreover, to illustrate that improper normalization (i.e. $\tau = 1$) can diminish the performance for both CCT and CED, observe that the RMSE simulation on the test set(s) is 2.0 and 0.3 [ticks/s] for $\Delta t/\tau = \Delta t = 4$ s and $\Delta t/\tau = \Delta t = 0.02$ s respectively.

6 Conclusion

In this paper, we have introduced the CT subspace encoder approach to identify nonlinear dynamical systems in the presence of latent states, external inputs, and measurement noise. We have shown that the proposed method can obtain highly accurate CT models only consisting of fully connected neural networks. The approach has improved computational cost and stability by considering multiple subsections where the initial state is estimated with an encoder function and by a state-derivative normalization term to improve optimization stability. We provided multiple theoretical proofs which provide additional insight and motivation for the method. These proofs are, increased cost function smoothness, necessary conditions for the existence of the encoder function and that for proper normalization in CT modeling one requires the state-derivative normalization term. Furthermore, we obtain state-of-the-art results on all three considered benchmarks.

7 REPRODUCIBILITY STATEMENT

- **Datasets:** *(i)* The CCT dataset is described in Schoukens & Noël (2017); Schoukens et al. (2017) and is available for download at `https://data.4tu.nl/articles/dataset/Cascaded_Tanks_Benchmark_Combining_Soft_and_Hard_Nonlinearities/12960104`, *(ii)* the CED dataset is described in Wigren & Schoukens (2017) and is available for download at `http://www.it.uu.se/research/publications/reports/2017-024/`, *(iii)* the EMPS datset is described in (Janot et al., 2019) and is available for download at `https://www.nonlinearbenchmark.org/benchmarks/emps`.
- **Code:** Both the implementation and experiments of CT SUBNET and neural ODE are available at *"Github link removed for double-blind reviews, see zip file from supplementary materials until de-anonymization"*.
- **Hardware:** It takes about 15 minutes to estimate a single CT SUBNET model and 2 hours for a single neural ODE model on a consumer laptop. A notable exception is CT SUBNET for EMPS which took about 10 hours due to the increased size and difficulty of the dataset.

REFERENCES

Helon Vicente Hultmann Ayala, Luciano F da Cruz, Roberto Z Freire, and Leandro dos Santos Coelho. Cascaded free search differential evolution applied to nonlinear system identification based on correlation functions and neural networks. In *2014 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pp. 1–7. IEEE, 2014.

Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. Learning dynamical systems from partial observations. *Second Workshop on Machine Learning and the Physical Sciences, NeurIPS*, 2019.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Gerben I. Beintema, Roland Toth, and Maarten Schoukens. Nonlinear state-space identification using deep encoder networks. In *Learning for Dynamics and Control*, pp. 241–250. PMLR, 2021.

Georgios Birpoutsoukis, Péter Zoltán Csurcsia, and Johan Schoukens. Efficient multidimensional regularization for volterra series estimation. *Mechanical Systems and Signal Processing*, 104: 896–914, 2018.

Julien Brajard, Alberto Carrassi, Marc Bocquet, and Laurent Bertino. Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model. *Journal of Computational Science*, 44:101171, 2020.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Charles K Chui, Guanrong Chen, et al. *Kalman filtering*. Springer, 2017.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *ICLR Workshop*, 2020.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(6):741–760, 2021.

Marco Forgione and Dario Piga. Continuous-time system identification with neural networks: model structures and fitting criteria. *European Journal of Control*, 59:69–81, 2021a.

Marco Forgione and Dario Piga. dynonet: A neural network architecture for learning dynamical systems. *International Journal of Adaptive Control and Signal Processing*, 35(4):612–626, 2021b.

Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *ICLR*, 2019.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Alexandre Janot, Maxime Gautier, and Mathieu Brunot. Data set and reference models of EMPS. In *Nonlinear System Identification Benchmarks*, 2019.

Ridvan Karagoz and Kim Batselier. Nonlinear system identification with regularized tensor network b-splines. *Automatica*, 122:109300, 2020.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Tohru Katayama. *Subspace methods for system identification*, volume 1. Springer, 2005.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR, 1999. ISBN 9780136566953.

Romit Maulik, Arvind Mohan, Bethany Lusch, Sandeep Madireddy, Prasanna Balaprakash, and Daniel Livescu. Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D: Nonlinear Phenomena*, 405:132368, 2020.

Francis J Murray and Kenneth S Miller. *Existence theorems for ordinary differential equations*. Courier Corporation, 2013.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Rishi Relan, Koen Tiels, Anna Marconato, and Johan Schoukens. An unstructured flexible nonlinear model for the cascaded water-tanks benchmark. *IFAC-PapersOnLine*, 50(1):452–457, 2017.

Antônio H Ribeiro, Koen Tiels, Jack Umenberger, Thomas B Schön, and Luis A Aguirre. On the smoothness of nonlinear system identification. *Automatica*, 121:109158, 2020.

Tim Rogers, Geoffrey Holmes, Elizabeth Cross, and Keith Worden. On a grey box modelling framework for nonlinear system identification. In *Special Topics in Structural Dynamics, Volume 6*, pp. 167–178. Springer, 2017.

Samuel H Rudy, J Nathan Kutz, and Steven L Brunton. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *Journal of Computational Physics*, 396:483–506, 2019.

Farnaz Sabahi and Mohammad Reza Akbarzadeh-T. Extended fuzzy logic: Sets and systems. *IEEE Transactions on Fuzzy Systems*, 24(3):530–543, 2015.

Michele Scarpiniti, Danilo Comminiello, Raffaele Parisi, and Aurelio Uncini. Novel cascade spline architectures for the identification of nonlinear systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(7):1825–1835, 2015.

Maarten Schoukens and Jean Philippe Noël. Three benchmarks addressing open challenges in nonlinear system identification. *20th World Congress of the International Federation of Automatic Control*, 50:446–451, 2017. doi: 10.1016/j.ifacol.2017.08.071.

Maarten Schoukens, P Mattson, Torbjörn Wigren, and Jean-Philippe Noël. Cascaded tanks benchmark combining soft and hard nonlinearities. In *4TU.ResearchData, Dataset*, 2017. doi: 10.4121/12960104.

Andreas Svensson and Thomas B Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

Jonas Weigand, Michael Deflorian, and Martin Ruskowski. Input-to-state stability for system identification with continuous-time runge–kutta neural networks. *International Journal of Control*, pp. 1–17, 2021.

Torbjörn Wigren and Maarten Schoukens. *Coupled electric drives data set and reference models*. Department of Information Technology, Uppsala Universitet, 2017.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: Learning hamiltonian dynamics with control. *ICLR*, 2020.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021a.

Hongpeng Zhou, Chahine Ibrahim, Wei Xing Zheng, and Wei Pan. Sparse Bayesian deep learning for dynamic system identification. *arXiv preprint arXiv:2107.12910*, 2021b.

8 Appendix

8.1 Proof of theorem 1

Recall that the subspace encoder loss function as in Eq. 3 can be expressed in the following form

$$\begin{aligned} V^{\text{enc}}(\theta) = \quad & \frac{1}{N-T-\max(n_a,n_b)+1} \sum_{n=\max(n_a,n_b)}^{N-T} \frac{1}{T}\sum_{k=0}^{T-1} \|y_{n+k} - \hat{y}_{n+k|n}\|_2^2, \\ \text{with} \quad & \hat{y}_{n+k|n} = h_\theta(x_{n+k|n}) = h_\theta(x((n+k)\Delta t|n\Delta t)), \\ & \dot{x}(t|n\Delta t) = \frac{1}{\tau} f_\theta(x(t|n\Delta t), u(t)) \\ & x(n\Delta t|n\Delta t) = \psi_\theta(u_{n-1},...,u_{n-n_b}, y_{n-1},...,y_{n-n_a}). \end{aligned} \tag{9}$$

Our aim is to derive the scaling in T of the Lipschitz constant L_{enc} as defined in

$$|V^{\text{enc}}(\theta_1) - V^{\text{enc}}(\theta_2)|^2 \leq L_{\text{enc}}^2 \|\theta_1 - \theta_2\|_2^2, \quad \forall \theta_1, \theta_2 \in \Theta \subset \mathbb{R}^{n_\theta}. \tag{10}$$

We aim to express L_{enc} in terms of the following Lipschitz constants

$$\|h_{\theta_1}(x_1) - h_{\theta_2}(x_2)\|_2^2 \leq L_h^2(\|x_1 - x_2\|_2^2 + \|\theta_1 - \theta_2\|_2^2). \tag{11a}$$

$$\|f_{\theta_1}(x_1, u) - f_{\theta_2}(x_2, u)\|_2^2 \leq L_f^2(\|x_1 - x_2\|_2^2 + \|\theta_1 - \theta_2\|_2^2), \tag{11b}$$

$$\|\psi_{\theta_1}(u_{\text{past}}, y_{\text{past}}) - \psi_{\theta_2}(u_{\text{past}}, y_{\text{past}})\|_2^2 \leq L_\psi^2 \|\theta_1 - \theta_2\|_2^2. \tag{11c}$$

for all $x_1, x_2 \in X \subset \mathbb{R}^{n_x}$ and $\forall \theta_1, \theta_2 \in \Theta \subset$. A sufficient condition for these Lipschitz constants to be finite is that the derivatives are finite on a compact set of inputs which is often the case in feed-forward neural networks.

Furthermore, to derive the scaling of L_{enc} we use known properties of the Lipschitz constants;

- The sum property: $c(x) = a(x) + b(x)$ has a Lipschitz constant of $L_c = L_a + L_b$.
- The multiplication property: $c(x) = a(x)b(x)$ has a Lipschitz $L_c = M_a L_b + M_b L_a$ where M_a is the maximal value of a on a closed set of inputs $x \in X$ and M_b being similarly defined.

Since the encoder loss function as in Eq. 3 can be written as a sum:

$$V^{\text{enc}}(\theta) = \frac{1}{N - T - \max(n_a, n_b) + 1} \sum_{n=\max(n_a,n_b)}^{N-T} V^{\text{sec}}(n, \theta_1) \tag{12a}$$

$$V^{\text{sec}}(n, \theta_1) = \frac{1}{T} \sum_{k=0}^{T-1} \|y_{n+k} - \hat{y}_{n+k|n}\|_2^2 \tag{12b}$$

$$\tag{12c}$$

where

$$|V^{\text{sec}}(n, \theta_1) - V^{\text{sec}}(n, \theta_2)|^2 \leq L_{\text{sec}}^2 \|\theta_1 - \theta_2\|_2^2, \tag{12d}$$

Then by the sum property this implies that $L_{\text{enc}} = L_{\text{sec}}$ since

$$L_{\text{enc}} = \frac{1}{N - T - \max(n_a, n_b) + 1} \sum_{n=\max(n_a,n_b)}^{N-T} L_{\text{sec}}. \tag{13}$$

Hence, it is sufficient to consider only a single subsection. Take $n = 0$ and drop the bar notation for simplicity. By the sum and multiplication properties, we derive get

$$|V^{\text{sec}}(\theta_1) - V^{\text{sec}}(\theta_2)| \leq 2/T \sum_{k=0}^{T-1} (M_y + M_k) \|\hat{y}_{1,k} - \hat{y}_{2,k}\|_2, \tag{14}$$

where M_y is the bound on $\|y(t)\|_2$ assuming a stable system and M_k, the bound on $\|\hat{y}_k\|_2$. The M_k bound scales the same as $\|\hat{y}_{1,k} - \hat{y}_{2,k}\|_2$ as shown in Ribeiro et al. (2020). The $\|\hat{y}_{1,k} - \hat{y}_{2,k}\|_2$ expression can be expanded by using Eq. 11a as

$$\|\hat{y}_{1,k} - \hat{y}_{2,k}\|_2^2 \leq L_h^2 (\|x_1(k\Delta t) - x_2(k\Delta t)\|_2^2 + \|\theta_1 - \theta_2\|_2^2). \tag{15}$$

Next, we aim to derive an expression for the Lipschitz constant $L_x(t)$ given in terms of

$$\|x_1(t) - x_2(t)\|_2^2 \leq L_x(t)^2 \|\theta_1 - \theta_2\|_2^2. \tag{16}$$

whereby Eq. (11c)

$$L_x(0) = L_\psi. \tag{17}$$

By considering a small increment in time of length h, we can express $L_x(t + h)$ in terms of $L_x(t)$. Using the fact that h is small we can use an Eurler step and discard higher order terms of h as;

$$\begin{aligned}
\|x_1(t+h) - x_2(t+h)\|_2^2 &= \|(x_1(t) - x_2(t)) + h/\tau(f_{\theta_1}(x_1(t), u(t)) - f_{\theta_2}(x_2(t), u(t)))\|_2^2 \\
&\leq \|x_1(t) - x_2(t)\|_2^2 + 2h/\tau \|x_1(t) - x_2(t)\|_2 \|f_{\theta_1}(x_1(t), u(t)) - f_{\theta_2}(x_2(t), u(t))\|_2
\end{aligned}$$

by the triangle inequality. Next, we can replace all the f terms by Eq. (11b) and x terms by Eq. (16) to derive an expression for $L_x(t + h)$ as

$$\begin{aligned}
&\leq \|x_1(t) - x_2(t)\|_2^2 + 2h/\tau \|x_1(t) - x_2(t)\|_2 L_f \sqrt{\|x_1(t) - x_2(t)\|_2^2 + \|\theta_1 - \theta_2\|_2^2} \\
&\leq \left(L_x(t)^2 + 2h/\tau L_x(t) L_f \sqrt{L_x(t)^2 + 1} \right) \|\theta_1(t) - \theta_2(t)\|_2^2 \\
&L_x(t+h) = \sqrt{L_x(t)^2 + 2h L_x(t) \sqrt{L_x(t)^2 + 1} L_f / \tau}.
\end{aligned}$$

This expression allows us to derive that the derivative of $L_x(t)$ is given by

$$\dot{L}_x(t) = \lim_{h \to 0} \frac{L_x(t+h) - L_x(t)}{h}, \tag{18}$$

$$= \sqrt{1 + L_x(t)^2} L_f / \tau. \tag{19}$$

which suggests that $\dot{L}_x(t)$ is continuous in t and has the closed form solution as

$$L_x(t) = L_x(0) + \int_0^t \dot{L}_x(t')dt' \tag{20}$$

$$= \sinh(tL_f/\tau + \text{arcsinh}(L_\psi)) \tag{21}$$

Now by substituting Eq. (21) into, (16), (15), (14) and using (13) we arrive at the following expression for L_{enc} as

$$L_{\text{enc}} = 2/T \sum_k (M_y + M_k) L_h(\sinh(k\Delta t L_f/\tau + \text{arcsinh}(L_\psi)) + 1) \tag{22a}$$

which scales in the limit of large T as

$$L_{\text{enc}} = \mathcal{O}(\exp(2T\Delta t L_f/\tau)), \tag{23a}$$

since $L_f > 0$ and $\Delta t > 0$. Note that the 2 in the exponent is from the multiplication with M_k which also scales as the y term as previously mentioned.

Furthermore, this bound cannot be lowered since the linear system $\dot{x}(t) = x(t)L_f/\tau$ already results in the scaling of $L_{\text{enc}} \sim \exp(2T\Delta t L_f/\tau)$.

8.2 RECONSTRUCTABILITY OF THE INITIAL STATE FROM PAST INPUT AND OUTPUTS

To derive the conditions on the existence of the encoder function suppose that we have a system given by

$$\dot{x}(t) = f(x(t), u(t)), \tag{24a}$$

$$y_n = h(x_n) + w_n, \tag{24b}$$

where $x_n = x(n\Delta t)$. For this system, if a state is given $x(t_0)$ along the input trajectory $u(t)$ one would in principle be able to compute $x(t)$ for all $t > t_0$. However, since we aim to construct the state given past outputs we need $x(t)$ for $t < t_0$ which requires backward in time integration. This backward integration on f is guaranteed to be unique if f is Lipschitz continuous in x for all u as by the Picard–Lindelöf theorem (Murray & Miller, 2013). Hence, since f is assumed to be Lipschitz continuous we can construct an operator f_d which can integrate backwards or forwards as

$$x_{n+1} = f_d(x_n, u_n) \rightarrow x_{n-1} = f_d^{-1}(x_n, u_{n-1}) \tag{25}$$

where u is subject to ZOH.

This operator allows us to construct past outputs as

$$\begin{aligned} y_{n-1} &= (h \circ f_d^{-1})(x_n, u_{n-1}) + w_{n-1} \\ y_{n-2} &= (h \circ f_d^{-2})(x_n, u_{n-1}, u_{n-2}) + w_{n-2} \\ &\vdots \\ y_{n-z} &= (h \circ f_d^{-z})(x_n, u_{n-1}, u_{n-2}, ..., u_{n-z}) + w_{n-z} \\ Y_n^{-z} &= (H \circ F_d^{-z})(x_n, U_n^{-z}) + W_n^{-z} \end{aligned} \tag{26}$$

where

$$f_d^{-p}(x_n, u_{n-1}, ..., u_{n-p}) = f_d^{-p+1}(f_d^{-1}(x_n, u_{n-1}), u_{n-2}, ..., u_{n-p})$$

is the application of f_d^{-1}, p times to obtain x_{n-p}. Furthermore, $Y_n^{-z} = [y_{n-1}^\top, y_{n-2}^\top, ..., y_{n-z}^\top]^\top$ and, $(H \circ F_d^{-z})$ with W_n^{-z} are similarly defined. To construct the initial state x_n, we need to invert Eq. (26). This inverse is also known as a reconstructability map (Katayama, 2005). For the inverse to exist, several necessary requirements can be given. One such necessary requirement is that a small perturbation to a solution x_n should change $(H \circ F_d^{-z})$, otherwise these solutions are indistinguishable from the output. This is formalized stating by that the matrix $\frac{\partial (H \circ F_d^{-z})(x_n, U_n^{-z})}{\partial x_n}$ has a null space of rank zero which is also known as the local observability condition. This is the same as the condition that the column rank of this matrix $\geq n_x$. A necessary requirement for this

column rank condition is that the number of columns is equal to or greater than the number of rows i.e. $zn_y \geq n_x$.

Hence, under the right conditions, it might be possible to solve Eq. (26) for a singular x_n since this equation is a nonlinear fixed point problem if W_n^{-z} is known. For the case that W_n^{-z} is unknown one can estimate the state $\hat{x}_n \approx x_n$ by solving the nonlinear regression problem;

$$\hat{x}_n = \arg\min_{\hat{x}_n} \|Y_n^{-z} - (H \circ F_d^{-z})(\hat{x}_n, U_n^{-z})\|_2, \tag{27a}$$

$$= \arg\min_{\hat{x}_n} \|L(\hat{x}_n, Y_n^{-z}, U_n^{-z})\|_2. \tag{27b}$$

Hence, both f uniformly Lipschitz continuous and $(\nabla_x L)^T \nabla_x L$ being full rank in $\hat{x}_n$ are necessary conditions for the existence of a unique reconstructability map.

Computing the reconstructability map for our model thus requires solving an optimization problem that becomes computationally infeasible during training. Hence, the encoder function aims to approximate the solution to Problem (27).